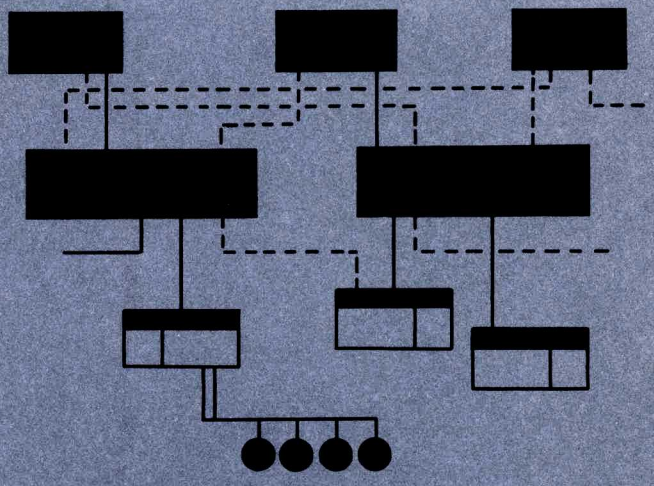
GE-635 System Manual



GENERAL ELECTRIC

CPB 371 A

GE-635

SYSTEM MANUAL

Rev. July 1964

GENERAL ELECTRIC

COMPUTER DEPARTMENT

PREFACE

This manual introduces the GE-625 and the GE-635 Information Processing Systems. All significant features and capabilities of the two systems are identical, except for memory access times and instruction execution times. In the interests of clarity and readability, this manual discusses the GE-635 only, which has a one microsecond memory cycle. However, all information (except timing data) is applicable to the GE-625, which has a two microsecond memory cycle. The Appendix includes a listing of all machine instructions with execution times for both the GE-625 and the GE-635.

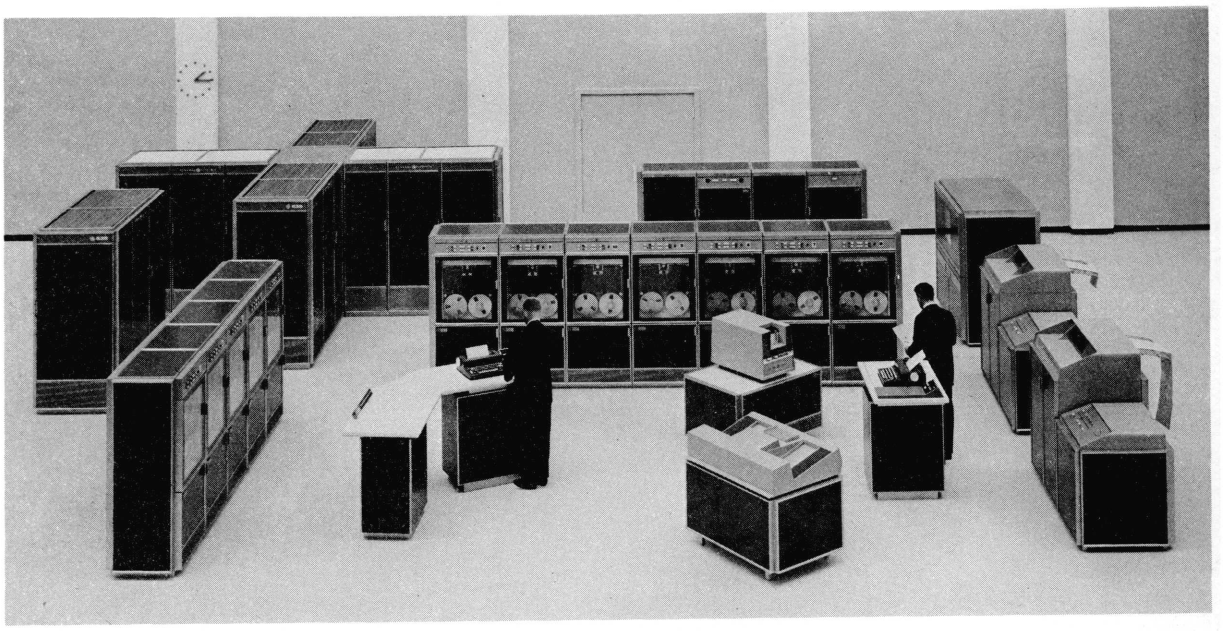
For more detailed information concerning either the GE-625 or the GE-635, refer to the GE-635 Programming Reference Manual, CPB-1004, and the GE-635 General Comprehensive Operating Supervisor Reference Manual, CPB-1002.

CONTENTS

VII. PERIPHERAL EQUIPMENT SUBSYSTEMS

APPENDIX

ILLUSTRATIONS



I. INTRODUCTION

The GE-635 Information Processing System is a modular, solid-state, digital computing system for large-scale business, scientific, military command and control and other real-time applications. The GE-635 equipment is complemented by a fully-integrated, yet modular, programming system that is controlled by the General Comprehensive Operating Supervisor (GECOS), an executive program that takes full advantage of the GE-635 equipment features.

The overall system design objectives for the GE-635 were to provide a user-oriented computation system with maximum reliability and the highest possible computation/cost ratio that would serve a user's present needs and could be expanded as his needs increased.

SYSTEM CONCEPTS

System objectives were met in design through careful attention to four concepts:

Modularity
Multiprogramming and multiprocessing capability
Integrated programming and operating system
Potential for growth

Modularity

Modularity provides full configuration flexibility. Basic equipment modules permit establishing a configuration for current needs that can be expanded to other configurations to meet growing needs. The basic modules, which provide increased reliability through provision for self-backup, are:

- Memory Modules--This module type contains a System Controller and one or two core storage units. All communication and control functions are routed through a System Controller that accesses the core storage units.
- Processor Modules--This module type performs the data movement, arithmetic, logic, comparison, and control operations. Processor instructions can address up to 262,144 consecutive words of memory.
- Input/Output Controller Modules--This module type is the coordinator of all input/output operations between Memory modules and the peripheral subsystems.

Modular input/output control capabilities of the GE-635 permit the use of a full range of peripheral equipment, including real-time, data communication, remote terminal, and mass storage devices, as well as conventional input/output devices.

Multiprogramming and Multiprocessing Capability

Multiprogramming is made possible through a combination of equipment and control program features.

The General Comprehensive Operating Supervisor (GECOS) is an operating system that adapts itself automatically to operate with any equipment configuration.

Dual mode operation restricts operating control of the multiprogramming environment to GECOS:

- Master mode, reserved for GECOS, allows unrestricted access to all of memory, permits input/output initiation through the Input/Output Controllers, and permits the setting of control registers.

- Slave mode, used by job programs and system programs such as FORTRAN, COBOL, Loader, Sort/Merge, etc., restricts address references to individual program boundary regions, and causes all memory references to be relative to the Base Address Register (BAR). The BAR is accessible only to control programs operating in the Master mode.

Dynamic relocation allows programs to be compacted in memory as individual programs are completed, allowing the system to use the released memory areas for new program allocation. In conjunction with dynamic relocation, these characteristics are significant:

- The Base Address Register is set by GECOS for each job program before that program gains control of the Processor.

- To insure GECOS control of the multiprogramming system, the programmer cannot set or restore the Base Address Register.

- Each programmer can look at memory as if his memory area always starts at zero, regardless of where his program ultimately resides in memory.

Memory protection insures complete program and data protection through program boundary limits imposed by the Processor and set by GECOS. Such boundary protection prevents job programs from erroneously affecting memory outside their assigned regions. Boundary protection for each program is automatically reset if the program is relocated during execution.

Effective interrupt orientation permits Processors and input/output devices to initiate interrupts whenever they require service from the operating system.

Integrated Programming and Operating System

The General Comprehensive Operating Supervisor provides complete control over activity allocation, execution, and termination of all programs. This centralized computer operating system, together with the memory protection and interrupt features, maximizes utilization of memory and input/output devices, enabling the user to realize rapid throughput.

Growth Potential

The user is assured that his GE-635 installation can grow as his needs increase. Modularity of design and a standardized memory interface permit additions of basic modules and peripheral devices as required.

Through data communications equipment, such as the DATANET-30, remote locations can be tied into the central computation system for remote inquiry and data processing, and multiple computer installations can communicate to form large computer complexes.

MODULARITY AND CONFIGURATIONS

A GE-635 system is implemented in a combination of the three basic modules. These basic modules can be arranged in a variety of configurations using the number of:

1. Memory modules needed to provide the required amount of storage
2. Processor modules needed to provide the required amount of computational ability, and
3. Input/Output Controller modules required for the complement of peripheral equipment included.

This modular construction results in computer configurations that are tailored to the precise needs of an installation. System growth is accomplished by adding the appropriate modules as they are needed. Added reliability is obtained because identical modules are employed in expanded systems. These identical modules provide backup for each other if individual modules malfunction. Figure I-1 illustrates a typical on-line configuration of a GE-635 utilizing one Processor module, one Memory module, and one Input/Output Controller module.

A system that uses multiple Processor modules is defined as a multiprocessor system. In such a system, only one Processor can be a Control Processor. All other Processors are subservient to the Control Processor. Subservient Processors can perform processing under the direction of the Control Processor; however, they cannot control system operations, nor can they directly initiate input/output transfers. The efficiency of multiprocessor configurations is inherently high. Figure I-2 illustrates a multiprocessor configuration with a real-time capability, using two Processor modules, three Memory modules, and two Input/Output Controller modules. In a multiprocessor system, the emphasis is on increased general-purpose system throughput.

A system that includes multiple Processor modules and multiple Memory modules contains switches that permit the establishment of either a multiprocessor system or a multicomputer system. In the multicomputer system, two or more Processors operate as Control Processors, each controlling its own Memory modules and initiating input/output operations through its own Input/Output Controllers. A system of this type can be regarded as two or more separate computer systems; however, the separate computers can share common Memory modules. A multicomputer configuration is illustrated in Figure I-3.

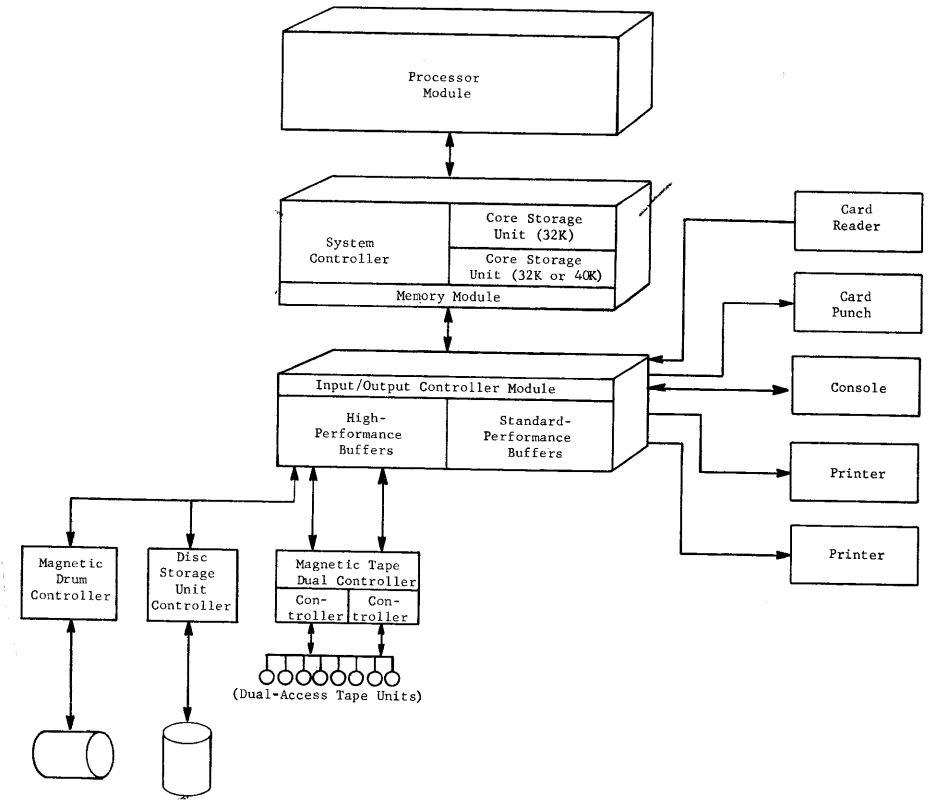


Figure I-1. On Line Configuration

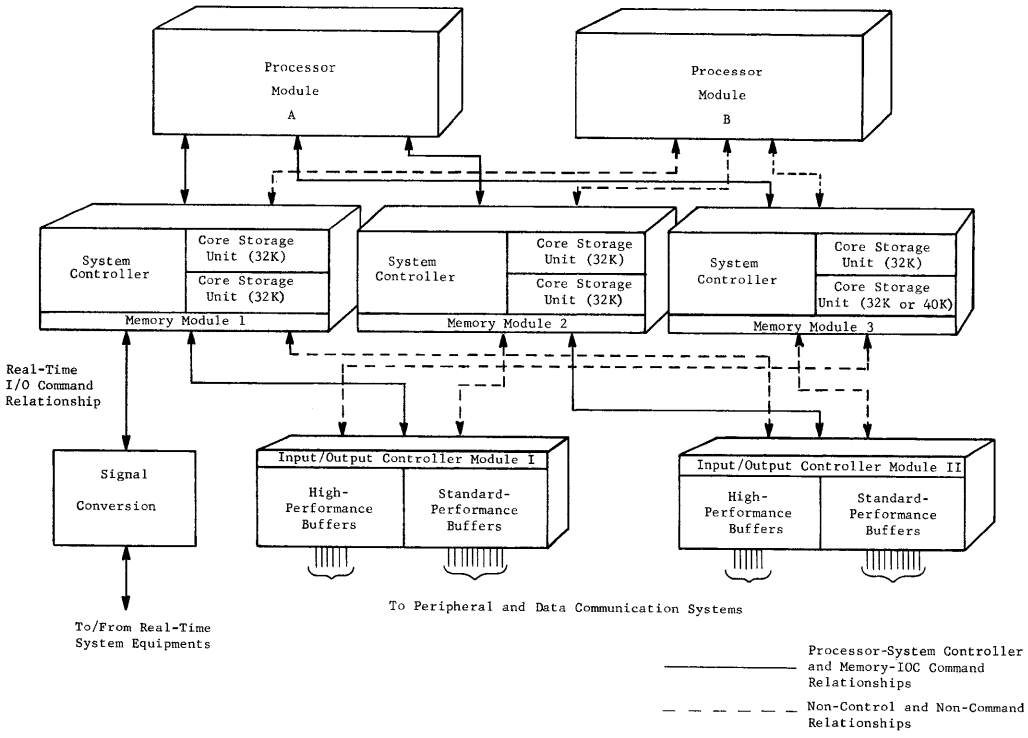


Figure I-2. Multiprocessor Configuration with Real-Time Capability

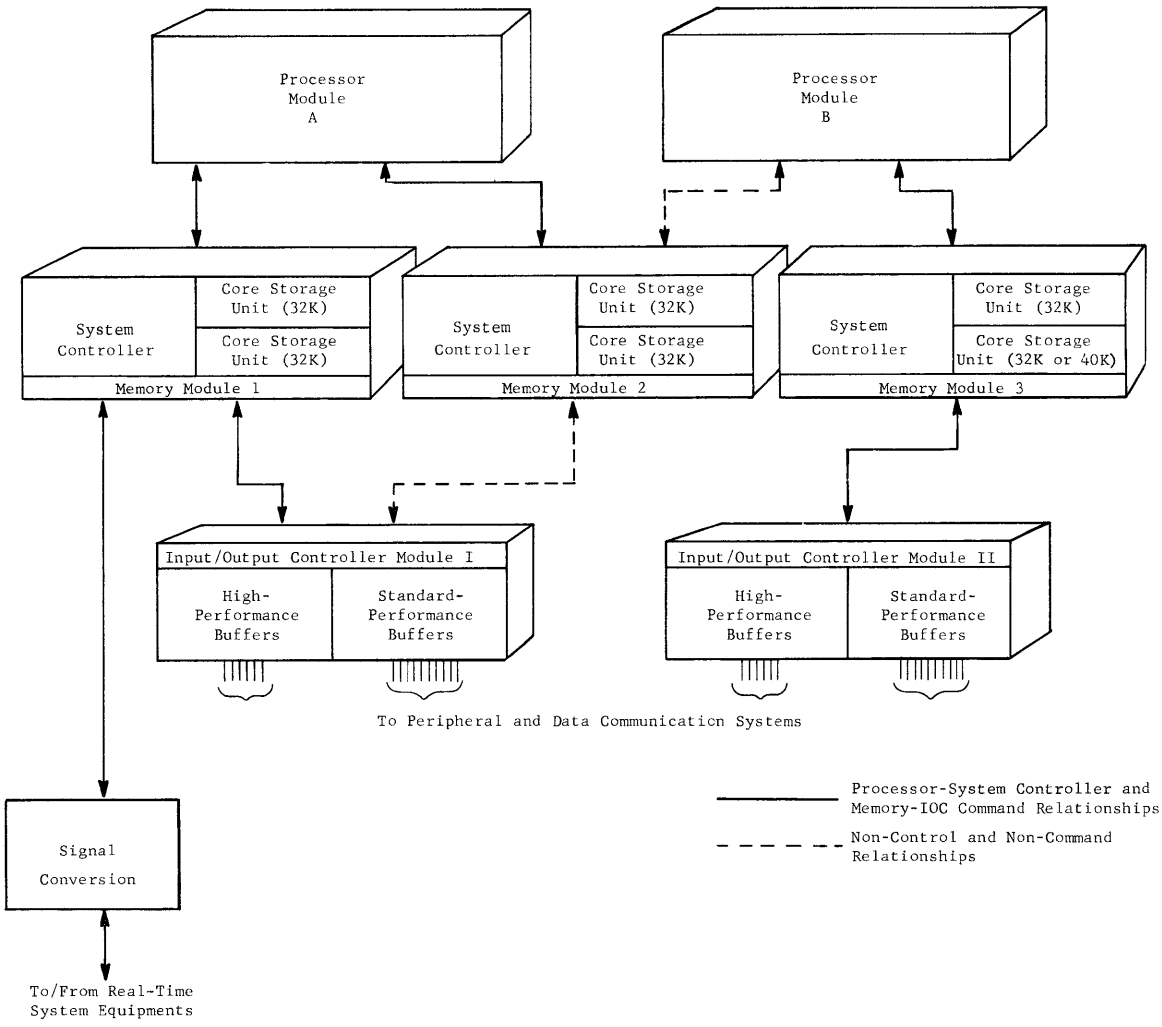


Figure I-3. Multicomputer Configuration

The maximum configuration for a GE-635 system in which each Processor and each Input/Output Controller has direct addressing of all associated Memory modules is limited to four System Controllers and associated core storage units. However, additional expansion of a multicomputer configuration can be accomplished if Processors communicate through shared Memory modules, rather than by direct addressing of each Memory module. The manner in which this can be done is shown in Figure I-4. Multicomputer configurations have enhanced reliability by permitting interchange of modules on a full system level as well as the individual module level.

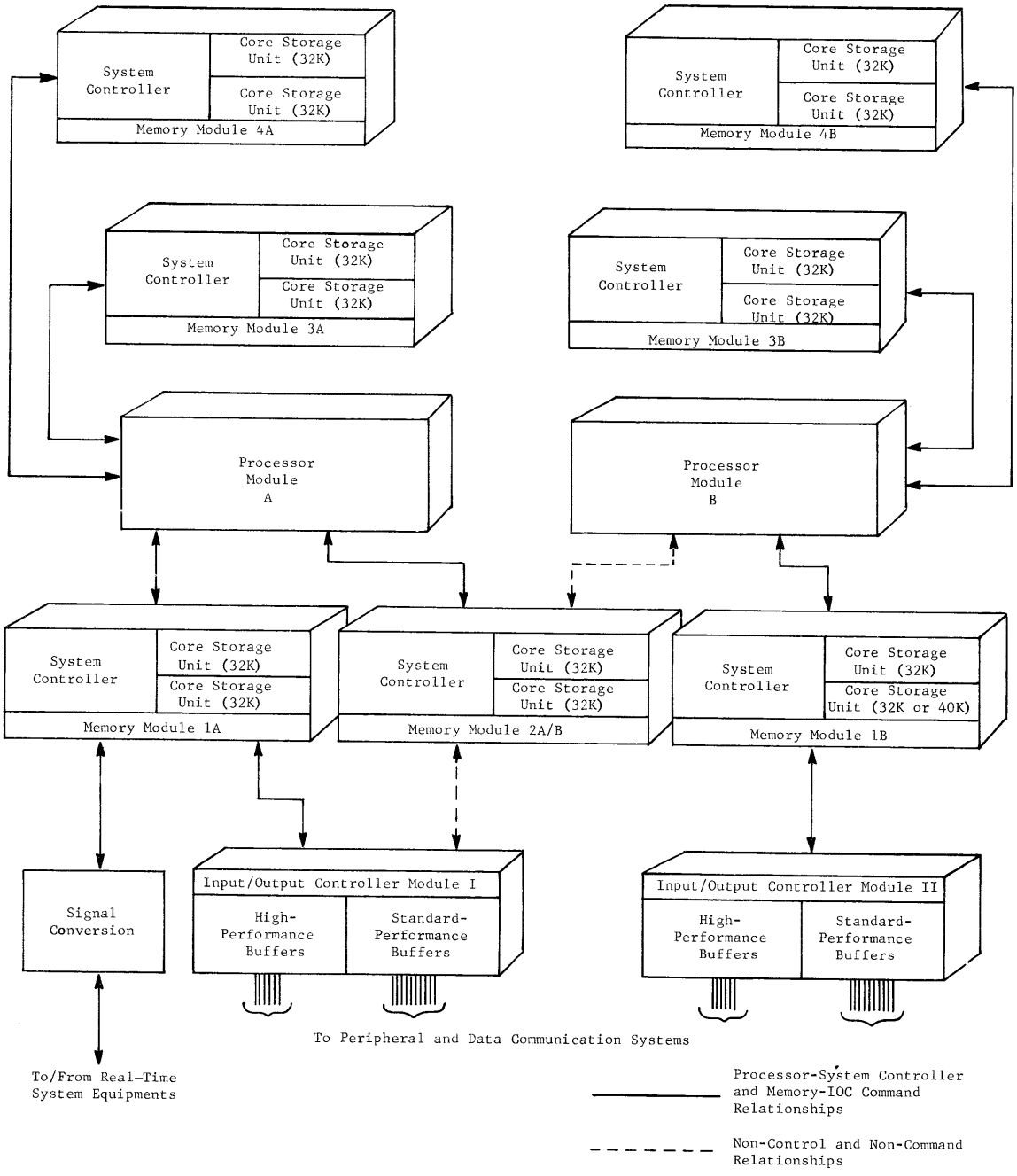


Figure I-4. Multicomputer Configuration with One Shared Memory Module

SYSTEM FEATURES

Certain GE-635 equipment features make significant contributions to the exceptional multiprogramming, high throughput, and rapid turn-around capabilities of the GE-635. These features, together with the integrated programming system under the control of GECOS, make possible the automatic supervision of the multiprogramming environment. These features permit GECOS to control:

Real-time interrupt response
Job scheduling
Hardware allocation
Input/output supervision
Local and remote media conversion
Routine housekeeping and accounting functions

Dual-Mode Operation

Dual mode operation is essential in a multiprogramming environment because only the operating system should be allowed to initiate input/output operations, set the interval timer, BAR, and other control registers. Without dual-mode operation, multiprogramming would require extreme discipline by the user to insure that he did not assume operating system functions.

Each Processor is capable of operating in two modes: the Slave mode or the Master mode. When a Processor is in the Master mode, no restrictions are imposed on a program. When a Processor is in the Slave mode, a program is inhibited from exercising a control relationship with Memory modules and associated peripheral devices. Also, while in the Slave mode, the Base Address Register (see below) controls relative addressing and memory limits protection for the program.

The General Comprehensive Operating Supervisor operates in the Master mode, whereas job programs operate in the Slave mode under control of GECOS. GECOS and job programs not currently being executed, together with their associated data, are protected from the job program being executed.

A Processor can be placed in the Slave mode at any time by program instruction. However, the Processor can be returned to the Master mode only by transfer to a legal system program entry from the job program, or by interrupt to a legal Master mode entry from a peripheral device. An attempt by a job program to access or transfer to any protected memory location at other than legal entry points, or to perform external control operations, results in an immediate fault routine. This feature provides positive protection for the entire system to allow it to operate in a multiprogramming-multiprocessing mode.

Dynamic Program Relocation

When the Processor is in the Slave mode, memory references are controlled by the Base Address Register (BAR). This register contains a base address associated with the job program that is automatically added to all job program addresses at execution time. The BAR also contains a designation of the number of 1024-word memory blocks that are assigned to the specified job program, thereby limiting the area of memory to which the job program has access. Thus, all job programs are stored in memory relative to a zero address and, each time memory is referenced, the base address is added to the address of the instruction to determine the actual address.

The BAR can be reset only in the Master mode. This restriction is essential for the operating system to be able to relocate job programs dynamically. If the programmer were able to access the BAR at any time, then programs could not be safely relocated. Master mode addressing ignores the contents of BAR and is absolute through the entire address range. The BAR permits entire job programs to be relocated in memory. This capability is desirable when many programs reside in memory and must be relocated frequently in order to consolidate memory areas that have been relinquished upon program terminations. Thus, it is also possible to interrupt a job program temporarily, place it in temporary storage, and later return it to any available block of memory locations without the need for software relocation.

Processor-Oriented Memory Protection

Two or more job programs simultaneously occupying the same Memory module must be protected from each other so that one program does not inadvertently store into the area outside its boundary limits. Each job program memory reference has the contents of the Base Address Register added to it, and a comparison is made to insure that this new address does not lie outside the allocated memory region. If it does, an out-of-bounds fault occurs.

Input/Output Controller-Oriented Memory Protection

Input/output activities requested by one job program must not be permitted to disturb any other job program unintentionally. Thus, each input/output channel on an Input/Output Controller has provisions for checking data transfers to and from memory to assure that preassigned data transfer area limits are not violated.

Interrupt Orientation

In an effective multiprogramming computer system, it is necessary to free both the equipment and the program from any specific timing requirements and from the burdens of checking other components of the system either for task completion or requests for service. Therefore, in the GE-635 system, devices that have completed tasks or that require service, generate interrupts to instruction processing. These interrupts can be generated by Processors as well as by input/output devices. Specific equipment features are included in the system to implement this interrupt-oriented philosophy.

Interval Timer

Each Processor contains a timer that initiates a program interrupt at the end of a variable interval. The timer can be loaded or set only in the Master mode. It consists of a 24-bit register that is automatically decremented every 16 microseconds. When the count reaches zero, the timer generates an interrupt, but continues to count. This interrupt is masked in the Master mode. The interval timer is used by GECOS for system control of the multiprogramming environment to time job programs for automatic job termination, to prevent a job program from monopolizing a Processor, and to provide accounting information by monitoring processing and input/output elapsed time.

Fault Interrupts

Because of the continuous access needs in a real-time multiprogramming application and in keeping with the interrupt-oriented philosophy of the GE-635 system, the Processor provides for continuous on-line operation by immediately responding to high-level interrupts called faults. Any operation that might otherwise cause the system to "hang up," causes a fault into

the Master mode control program automatically, so that immediate remedial action can be taken. Also, the system can call another job into execution if a current job encounters difficulty.

For each potential fault condition, two instructions in fixed memory locations are executed when the related fault occurs. The first instruction normally safe-stores the location counter and indicators; the second causes a transfer to the related fault routine. These fixed locations, reserved for Processor use only, are in the control supervisor area.

Fourteen conditions can cause faults:

- Exponent overflow
- Exponent underflow
- Arithmetic overflow
- Memory parity

} Maskable

- Connect (for Processor/Controller communication)
- Time runout
- Machine lockup
- Illegal operation code
- Command (Master/Slave mode violations)
- Operation not complete
- Divide check
- Memory address violation
- Shutdown
- Startup

} Non-maskable

Real-Time Interface

The GE-635 is designed to permit real-time devices to communicate directly with memory. The interface, together with memory, permits independent data movement and program initiation under complete control of the external device. Thus, the external device can operate in either the conventional processor-controlled mode or in a self-controlled mode which, for specialized applications, highly optimizes system performance.

INTEGRATED PROGRAMMING AND OPERATING SYSTEM

The basic features of the GE-635 system and the manner in which these features interact implies a fundamental requirement for control program integration with the equipment. This overall system integration is achieved through the General Comprehensive Operating Supervisor, GECOS, which is provided by General Electric.

The GE-635, regardless of configuration, functions entirely under the control of GECOS. The integrated programming system is implemented in GECOS-controlled modules, enabling the user to select only the required modules for a given application and add or delete modules as needs change.

GECOS utilizes the previously-discussed equipment features to form a comprehensive system for the supervision of a multiprogramming environment. It controls all activities, including real-time interrupt response, job scheduling, equipment allocation, input/output supervision, local and remote media conversion, and routine housekeeping and accounting functions.

The entire programming system reflects the design objectives of providing user-oriented software. Programming system characteristics include:

- GECOS (and other system software) operates from a mass random access storage base. For this reason, magnetic tapes are not required for GECOS Library files, FORTRAN compilation, or other system program storage or execution. It is also permissible for GE-635 users to store their job programs on the magnetic disc included in the GE-635 system. This mass random access capability provides for maximum system throughput and minimum job turn-around time.

- To further facilitate throughput and turn-around time, the entire software system, under control of GECOS, is interrupt-oriented. Multiple levels of input/output interrupts are immediately serviced on a priority basis.

- Symbolic file assignments provide complete flexibility of peripheral allocation for dynamic scheduling and system reconfiguration.

- Control programs are written in modular form, are well-documented, and can be readily modified to tailor the system to specific operating conventions and application requirements. Also, special-purpose real-time and remote input/output control software can easily be interfaced with GECOS.

- Programmers writing for the GE-635 system need not be concerned that their programs will be executed in a multiprogramming environment. The same conventions that free the programmer from concern about absolute memory and peripheral requirements also provide for system growth without reprogramming.

A more complete description of GECOS is contained in Chapter V, Programming Systems. The compilers, MACRO Assembler, utility programs and subroutines, and other programming aids included in the integrated programming system are also described in that chapter.

II. MEMORY MODULES

The Memory module is the hub for all information transferred between Processor modules and Input/Output Controller modules. The Memory module consists of a System Controller and either one or two associated core storage units. System Controllers are independent of each other and can function simultaneously, thus providing overlapped memory accesses in multiple-controller systems.

Normally, a memory size of 32k, 40k, 64k, or 72k words is associated with one System Controller. Each memory module represents a continuous portion of the total address range of associated modules. Only the highest-order Memory module can use the 40k or 72k memory size.

The System Controller has as many as eight channels for connection to Processor and Input/Output Controller modules, and it also contains memory protection logic and program-interrupt cells.

CHARACTERISTICS OF THE SYSTEM CONTROLLER

The System Controller performs most of the priority and control functions in the overall GE-635 system. These functions are:

- Control of communications between the system modules.
- Control of program interrupts for multiple-processor jobs and system programs, as well as Input/Output Controller requests for service.
- Protection of memory areas.
- Switching of control signals, addresses, and data to and from the Memory module.
- Provision for zone control (character handling).

The System Controller contains all the logic required for performing control functions and address checking. The controller also includes the following items:

- Eight memory ports for connection to the active modules (Processor, Input/Output Controllers, and other high-performance controllers and real-time devices). These ports are assigned a priority to permit the servicing of all demands in an orderly manner.
- A program-addressable memory port "lockout" mask that can be set by a control program to inhibit data transfers and interrupts.

- Sixteen program execute-interrupt cells. Any of the components interfacing with the System Controller can set the interrupt cells as required to initiate appropriate Control Processor activity.

- A program-addressable execute-interrupt mask. The control program can override the built-in priority of the interrupt cells and exercise complete control of all system interrupts as required.

- A Control-Processor-designation register. In a multiple-processor configuration it is desirable (for reliability and reconfiguration purposes) to designate different Processors alternately as the Control Processor for the system. The System Controller has an eight-position switch that corresponds to the eight channels to which the Processors and other active modules are connected. This switch is used to direct interrupt and control information to the appropriate Processor.

- A program-addressable memory protect register. In certain applications, it is advantageous to have a common data array that is not contiguously located with respect to the multiple-programs accessing it. A register in the System Controller can be set by the control program to protect various multiple, non-contiguous blocks of memory as required. The program-addressable memory protect register is a separate and independent feature that is unrelated to the standard Processor-oriented and Input/Output Controller-oriented memory protection.

REAL-TIME INTERFACE

The high-performance System Controller interface is derived from extensive experience with real-time military systems requirements. Its design provides maximum data handling capability with minimal interference with Processor activities.

The System Controller interface is basically bidirectional and word-oriented, and will accept a wide variety of real-time devices. It permits direct addressing of the memory locations involved without Processor intervention. This is of major importance with complex peripheral controllers requiring multiple queues, and for communication between a Processor or an Input/Output Controller and a Memory module.

The System Controller interface can also initiate an interrupt, specifying the interrupt cell to be used and the action to be taken by the interrupted Processor.

Real-time capabilities are further discussed in Chapter VI.

CORE STORAGE UNITS

Each Memory module contains either 32,768 or 65,536 words of magnetic core storage. Additionally, any one Memory module in a system contains an additional 8,192 words to accommodate the General Comprehensive Operating Supervisor (GECOS). Each word of core storage consists of 36 data bits, plus one parity bit. Although the single precision word size that is used throughout the system is 36 data bits, each access to memory, whether for data or instructions, obtains a double word of 72 data bits.

The total memory cycle time for each Memory module is defined as the time from the start of command execution to the completion of the read-write cycle. Defined in this manner, the GE-635 memory cycle time is 1 microsecond for a full 74-bit access (two words plus a parity bit for each word). No overlap of memory operation is considered in determining this time.

III. PROCESSOR MODULES

The Processor module has full program execution capability and conducts all actual computational processing within the GE-635 system. The Processor, which communicates only with System Controllers and associated memory, consists of an operations unit and a control unit. The operations unit contains the logic to execute arithmetic and logical operations. The control unit provides the interface between the operations unit and the System Controller. It also performs instruction fetching, address preparation, memory protection, and data fetching and storing. Both units operate with relative independence and maximum overlap to provide the highest rate of instruction execution.

SYSTEM CONFIGURATION FLEXIBILITY

The GE-635 Processor includes standard features that directly contribute to the formation of multiprocessor and multicomputer system configuration flexibility or immediate reconfiguration.

The Processor is equipped with as many as four channels for connection to Memory modules and can directly address 262,144 words of memory. All instructions utilize a full 18-bit address field. Memory addresses are assigned consecutively, beginning with zero and continuing through the full available memory for those systems with less than a full complement of memory. Each Processor maintenance panel provides manual switches for assigning any 32k memory block to any modular 32k address within the range of continuous memory.

For example, in a two-Processor, three-Memory system (such as that illustrated in Figure I-3), both Processors can share the low-order 32k of memory, but address the upper 32k in different Memory modules.

In addition to provision for reassignment of basic memory block addresses, it is possible to assign fixed memory locations in a manner that avoids conflict in a multiprocessor system. The Processor always assumes that these locations are in low-order memory. However, if a low-order memory is common to several Processors, these fixed memory locations would overlay each other. To prevent this, manual reassignment of these fixed locations within low-order memory is permitted so that locations are uniquely defined for each Processor. Control relationships are thus established so that all interrupts are answered by one Processor, and all special locations are in one memory bank. Maximum multiprocessor effectiveness is thus achieved by relieving the balance of memory of all encumbrances and resolving potential conflict between Processors.

REGISTER DESCRIPTIONS

The internal Processor registers that are accessible to the program are:

Name	Mnemonic	Length
Accumulator Register	AQ	72 bits
Eight Index Registers (n = 0, 1, ... , 7)	Xn	18 bits each
Exponent Register	E	8 bits
Base Address Register	BAR	18 bits
Indicator Register	IR	18 bits
Timer Register	TR	24 bits
Instruction Counter	IC	18 bits

- The AQ-Register can be used:

  In floating-point operations as a mantissa register for both single and double precision;

  In fixed-point operations as an operand register for double precision, and each half independently of the other as two operand registers for single precision; the halves become the A-Register (bits 0 through 35) and the Q-Register (bits 36 through 71); and

  In address modification each half of the A-Register and of the Q-Register can be the source of an index; these halves then become AU (bits 0 through 17), AL (bits 18 through 35), QU (bits 0 through 17), and QL (bits 18 through 35).

- The Xn-Registers can be used:

  In fixed-point operations as operand registers for half-precision;

  In address modification as sources of index quantities.

- The E-Register supplements the AQ-Register in floating-point operations as exponent register.

- The Base Address Register is used in address translation and protection. It denotes the base address and the number of 1024-word blocks assigned to the program being executed.

- The Indicator Register is a generic term for all of the program-accessible indicators within the Processor; the name is used where the set of indicators appears as a register, that is, as a source or destination of data. The format of the Indicator Register contents as they would be reflected in memory is shown in Figure III-1.

- The Timer Register is decremented by one each 16 microseconds and a Timer Runout fault occurs whenever its contents reach zero. If Timer Runout occurs in the Master mode, the fault does not occur until the Processor returns to the Slave mode.

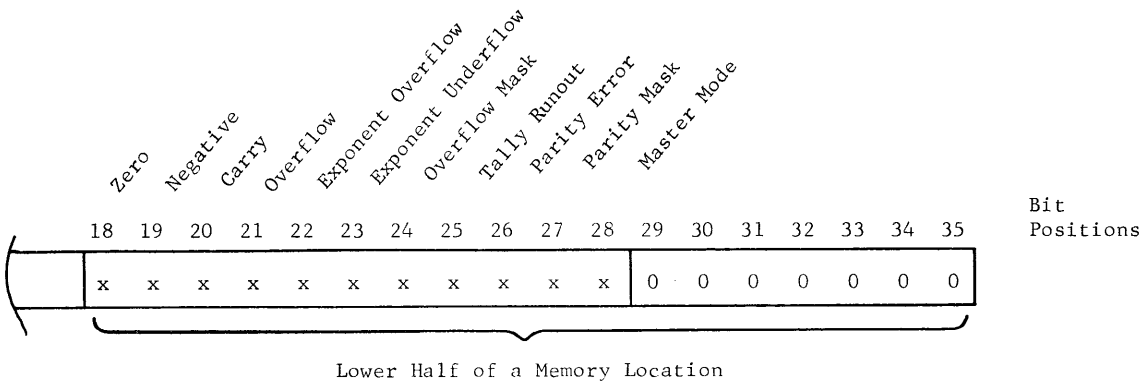


Figure III-1. Indicator Register Format

- The Instruction Counter stores the address of the next instruction to be executed.

WORD FORMATS

Fixed-Point Data

Fixed-point operations are conducted with 36-bit, single-precision operands; 72-bit double-precision operands; or 18-bit, half-word operands. The formats of the three types of operands are shown in Figure III-2.

Fixed-point data is represented in the two's complement number system. The range of single-precision numbers is $2^{35}-1$ to -2^{35}, the range of double-precision numbers is $2^{71}-1$ to -2^{71}, and the range of half-word operands is $2^{17}-1$ to -2^{17}.

Floating-Point Data

Floating-point operations are conducted with either 36-bit, single-precision operands or 72-bit, double-precision operands. The formats of each are illustrated in Figure III-3.

Floating-point numbers are represented by a mantissa of 28 or 64 bits and a binary exponent of 8 bits. Both the exponent and the mantissa are represented in the two's complement number system. The first bit of the mantissa indicates the sign of the quantity, and the exponent has a range of +127 to -128.

Alphanumeric Data

Six 6-bit characters are contained in each data word. The designation of the characters within the word is as shown in Figure III-4. Any combination of characters can be read or written to or from memory.

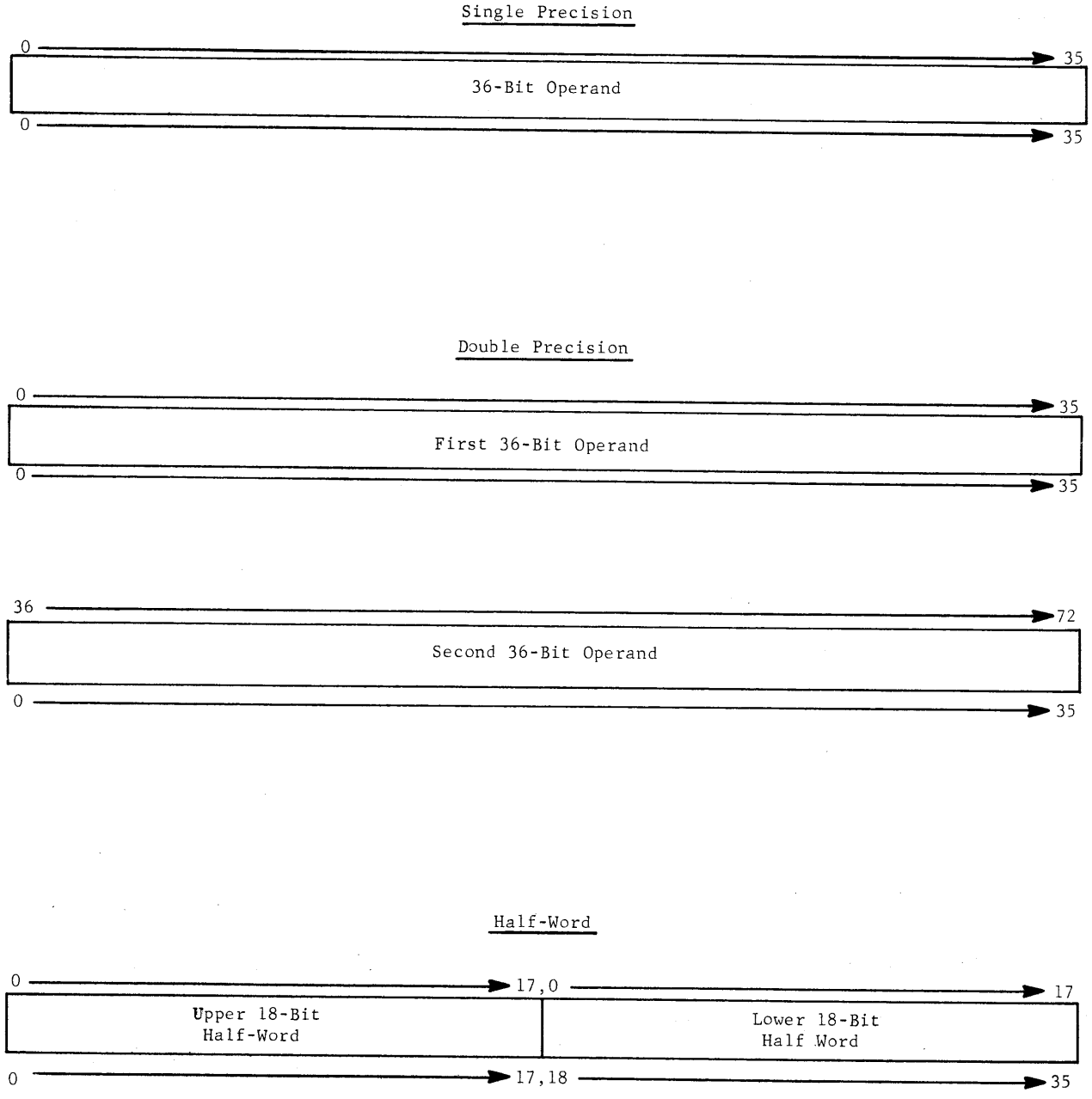


Figure III-2. Fixed-Point Data Formats

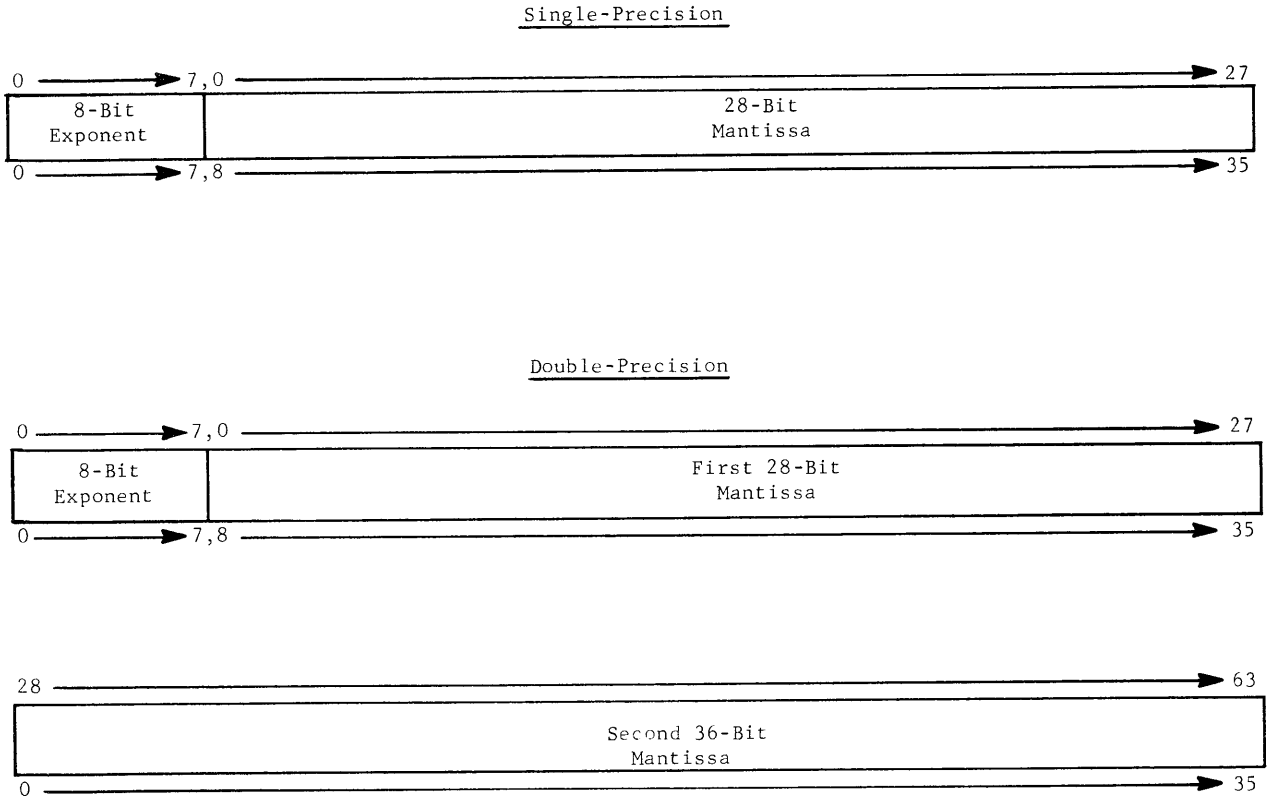


Figure III-3. Floating-Point Data Formats

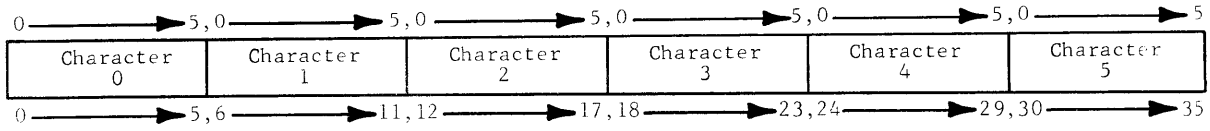


Figure III-4. Alphanumeric Data Format

Instruction Format

The instruction repertoire of the GE-635 Processor consists of 170 basic single-address instructions. With the exception of three Repeat instructions and the character-handling instructions. each instruction has the format shown in Figure III-5.

The operation (OP) part of the instruction specifies the operation to be performed and the registers that are involved.

The address (y) field of the instruction word specifies the storage location to obtain an operand or to place the result of the specified operation. Some of the exceptions to this general rule are:

1. Shift instruction, where the address field designates the number of bit positions to shift

2. Program sequence transfers, where the address field designates the location of the next instruction to be executed

3. A group of instructions where the address field specifies sub-operations.

The address modifier (t_m) portion of the tag field specifies how the address contained in the instruction word is to be modified to form the effective address Y of the resultant or the operand.

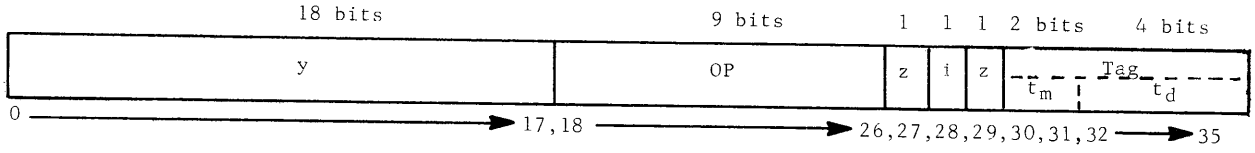


Figure III-5. General Instruction Format

The designator (t_d) portion of the tag field specifies the type of register modification to be used.

The interrupt inhibit (i) portion of the instruction word controls program interrupts in the Processor by either fault or program execute interrupts from the Memory.

The (z) fields must be zero, and are reserved for future compatibility with other General Electric 600-line systems.

ADDRESS MODIFICATION

Modification Types

The GE-635 performs address modification in four basic ways, using the identifiers R, RI, IR, and IT. The modification types R, RI, IR, or IT are specified by unique binary codes placed in the t_m field of the instruction word. The registers used to modify R, RI, or IR addresses are indicated in the t_d field of an instruction or indirect word.

1. Register modification (R)--Modify the address by adding to it the contents of the indicated register, producing the effective address of the operand. Indexing registers are X0-X7, AU, AL, QU, QL, and IC. The direct operand modifier symbols DU and DL may be used to treat the address directly as the operand.

2. Register then Indirect (RI)--First perform the indicated register modification of the address, as with R; then obtain an indirect word and conduct the modification specified in the indirect word.

3. Indirect then Register (IR)--First obtain the indirect word from the original address; then conduct the modification specified in the indirect word. Upon completion of the indirect addressing, perform the indicated register modification in the last IR word encountered. (Some special conditions for modification may be encountered in the indirect words; these involve R, RI, and IT.)

4. Indirect then Tally (IT)--Obtain the indirect word using the address modification type IT as specified in the t_m field of the instruction; then use the new address field of the indirect word to get the effective operand address. Next, follow one of nine possible variations, indicated in the t_d field of the instruction or indirect word that specified IT. (Two of the nine variations (SC and CI) are used for handling characters in the GE-635.)

The nine possible variations of the IT modification are listed and described below. When used, each variation mnemonic appears as a unique bit configuration in the t_d field of its instruction.

VARIATION SUMMARY UNDER IT MODIFICATION

Mnemonic	Name
(t_d) = I	Indirect (unmodified)
= ID	Increment address, decrement tally
= DI	Decrement address, increment tally
= SC	Sequence character
= CI	Character from indirect
= AD	Add delta
= F	Fault
= IDC	Increment address, decrement tally, and continue
= DIC	Decrement address, increment tally, and continue

IT Variation Descriptions

I — This tag is used to reference an indirect word without disturbing any part of the word. See Figure III-6.

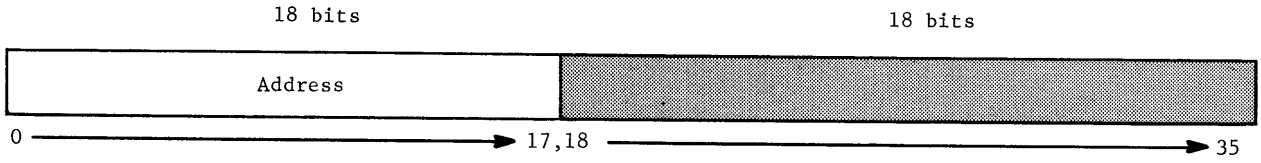


Figure III-6. Indirect Word as Interpreted Under I

ID — This tag is used to reference an indirect word (Figure III-7) and to increment the address by 1 and decrement the tally of the indirect word by 1.

DI — This tag is used to reference an indirect word and to decrement the address by 1 and increment the tally of the indirect word by 1.

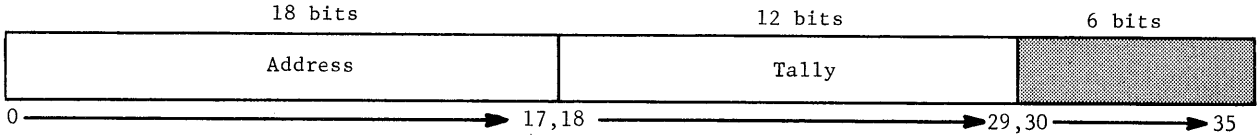


Figure III-7. Indirect Word as Interpreted under ID or DI

SC — This tag is used to reference an indirect word (Figure III-8) and to decrement the tally by 1, increment the character position by 1, and to specify the next character position--all in the indirect word.

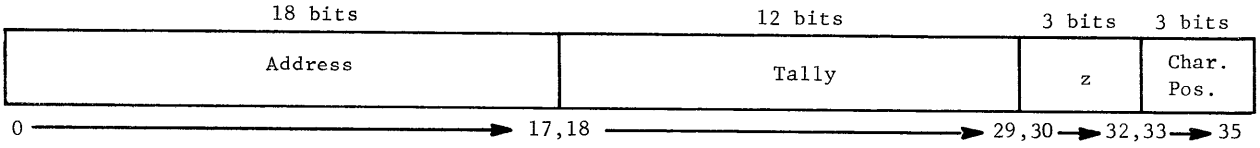


Figure III-8. Indirect Word as Interpreted under SC

CI — This tag is used to reference an indirect word (Figure III-9) and to operate on a particular character position without disturbing the indirect word.

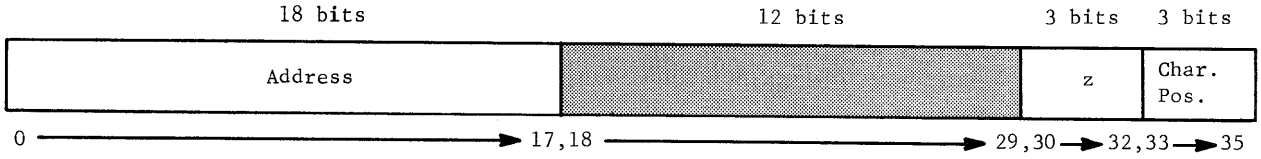


Figure III-9. Indirect Word as Interpreted under CI

AD — This tag is used to reference an indirect word (Figure III-10) and to increase the address field by the figure in the delta portion of the instruction. The tally portion of this indirect word is decremented by 1.

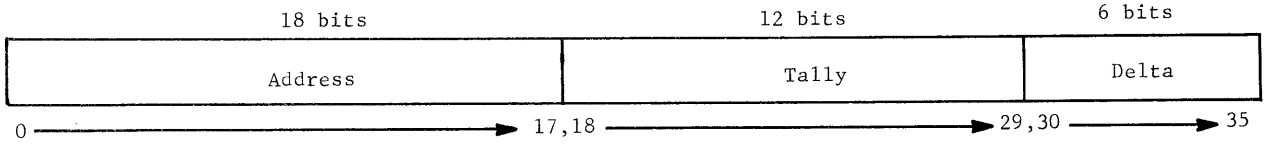


Figure III-10. Indirect Word as Interpreted under AD

F — The use of this address modification will cause a fault interrupt to occur.

IDC — This instruction modification is similar to ID modification, except that t_m and t_d fields specify further address modification. See Figure III-11 for indirect word interpretation.

DIC — This tag modification is similar to DI modification, except that the t_m and t_d fields specify further address modification.

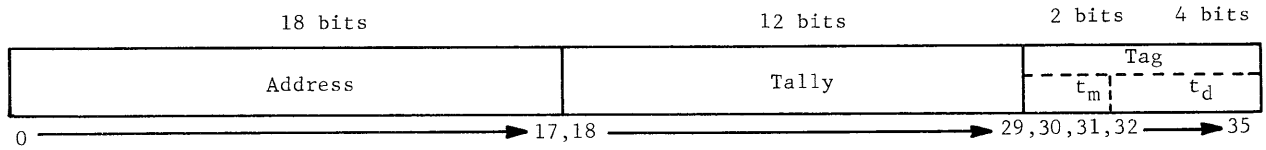


Figure III-11. Indirect Word as Interpreted under IDC and DIC

Address modification in the GE-635 using registers takes zero time; that is, under R, RI, IR modification, indexing the operand address of an instruction or indirect word adds no time to instruction execution. Under R modification, the DU and DL variations reduce normal execution time by 0.5 microseconds.

Character Operations

Operations involving characters are performed using instructions that specify the IT type of address modification in the t_m field. The I of IT obtains an indirect word that, in turn, references a location containing the character or characters to be used. The indirect word holds tallying and character identifying information while the referenced location holds the character data. The t_d field of the original instruction must contain the sequence character (SC) or character from indirect (CI) designator variations.

Each time the original instruction is used under the SC variation, the tally and character position information in the indirect word (see Figure III-8) are automatically changed by one 1) to prepare for operating on the next sequential character and 2) for terminating the SC operations when the tally reaches zero. Characters are numbered 0 through 5. After operating on character

number 5, the Processor automatically increases the data word address by one and starts again at character 0 in the new location.

When a single character within a word is to be used repeatedly with the IT modification, the t_d field designates CI. The indirect word format is that shown in Figure III-9. The data movement process parallels that for the SC variation but the indirect word is not altered.

For both variations, the character position field of the indirect word (bits 33 through 35) specifies in octal the character position of the memory location to be used in instruction execution. Figure III-12 illustrates typical single-character operations to and from memory, assuming a character position designation of 3.

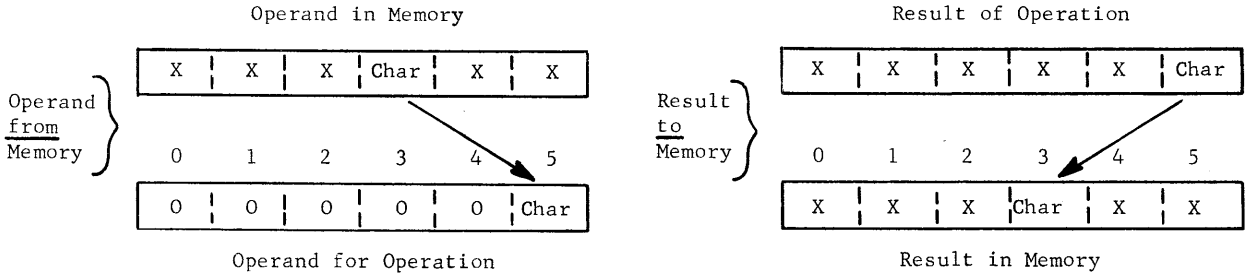


Figure III-12. Typical Character Operations

For operations where the operand is taken from memory, the effective operand is presented as a single word with the specified character right-justified to character position 5. The remaining positions (0 through 4 in Figure III-12) are presented as zeros.

For operations where the resultant is placed in memory, character 5 of the resultant replaces the specified character (3 in Figure III-12) in memory and the remaining character positions are not affected.

INSTRUCTION REPERTOIRE

Most of the instructions available on the GE-635 system are familiar to experienced programmers of large-scale computers. However, additional instructions have been provided to give the programmer extended capability for character handling, decision-making, and advanced programming techniques involving list processing.

The 170 basic instructions available are grouped as follows:

Data Movement and Shifting	36
Fixed-Point Arithmetic	34
Boolean Operations	21
Comparison	16
Floating-Point Arithmetic	30
Transfer Control	14
Special Operations	11
Master Mode Operations	8

These basic instructions have provision for multiple variations by permitting the programmer to specify not only the type of address modification desired, but also the source registers associated with particular operation codes. For example, the operation field for a Transfer and Save Instruction Counter in Index instruction specifies the index in the operation field, leaving full address modification capability free for destination calculation.

A Processor module permits efficient operations to be performed on 18-, 36-, and 72-bit operands. The 18-bit operands are stored with, and fetched as, the left half of the instruction.

The following paragraphs briefly describe salient features of the major instruction types. A complete listing of the full instruction repertoire and execution times is included in the Appendix.

Data Movement

Character handling and manipulation is facilitated by the "indirect and tally" indexing option, and by instructions for directly loading and storing selected characters of the accumulator or quotient register.

Instructions are also included for directly loading the index registers from either memory or the A/Q-Register, directly storing any register into memory, and for loading registers with the complement of the memory location specified.

Shifting

Shifting is accomplished using a "gear-shifting" algorithm, so that long shifts are executed essentially as fast as short shifts. The A- and Q-Registers can be shifted individually or as one unit. The Shift commands include right or left shift arithmetic, right shift logical, and left shift rotate (right shift rotate was omitted because the high speed of the left shift rotate makes the right shift rotate unnecessary).

Fixed-Point Arithmetic

Fractional and integer instructions for both multiplication and division afford the programmer freedom from scaling the results of such operations. Fractional multiplications are performed with the multiplicand in the A-Register; the result appears in bit positions 0 through 70 of the AQ-Register, automatically scaled with the binary point to the left of position 0. Integer multiplications are performed with the multiplicand in the Q-Register; the result appears in bit positions 1 through 71 of the AQ-Register, automatically scaled with the binary point to the right of position 71.

Fractional divisions use the full range of the AQ-Register for the dividend; the quotient appears in the A-Register with the remainder in the Q-Register. The binary point is automatically scaled to the left of position 0. Integer divisions have the integer dividend in the Q-Register, with the binary point to the right of position 35. After division, the quotient is in the Q-Register with the binary point automatically placed to the right of position 35, and the remainder is in the A-Register.

Normally, integer operations of divide and multiply occur in the Q-Register, and fractional operations of divide and multiply occur in the A-Register. This convention permits easy programming of fixed-point arithmetic operations.

Two arithmetic and three logic instructions used frequently in program coding are provided for combining the contents of memory locations directly with the contents of registers and storing the results in the same locations, without recourse to separate Store instructions. In all cases, the programmer can use the 18-bit indexing registers, X0 through X7, and the 36-bit A- and Q-Registers. In effect, the Add and Subtract to Storage instructions make arithmetic accumulators of all available memory locations. In all cases, the register contents are undisturbed.

Boolean Operations

The logical operations AND, Exclusive OR, and Inclusive OR are permitted between storage and the Xn-Registers, the A- and Q-Registers, and the AQ-Register.

Comparison

Compare operations do not alter the contents of storage or the specified register, but merely set or clear the appropriate indicator as the result dictates. The compare instructions enable the programmer to make many types of program decisions.

The fixed-point compare instructions are:

Instruction	Registers Used	Principle Functions
Compare Magnitude	A	Compare absolute values
Compare with Register	A, Q, AQ, X0-X7	(1) Compares algebraic values (2) Compares characters
Comparative AND with Register	A, Q, AQ, X0-X7	Tests for all zeros in word fields
Comparative Not-AND with Register	A, Q, AQ, X0-X7	Tests for all 1's in word fields
Compare Masked	A	Searches for identical, selectable word fields
Compare with Limits	A, Q	Searches for a value within selectable limits.

Floating-point compare instructions are included for single- and double-precision operations on absolute values and algebraic values. All compare instructions are repeatable using the RPT, RPD, or RPL instructions described later under the heading, Special Operations.

Floating-Point Arithmetic

Floating-point operations can be performed on both single- and double-precision data words; complete sets of data movement, arithmetic, and control instructions are provided for use in both types of operations. Unless otherwise specified by the programmer, the mantissas of all floating-point operation results are automatically normalized by the hardware. In additions and subtractions, addends and subtrahends are automatically aligned.

Operations on floating-point numbers are performed using an extended register composed of a 72-bit AQ-Register, which holds the mantissa, and a separate 8-bit exponent register; operations on the exponent and mantissa are performed by two separate adders. The existence of separate exponent and mantissa registers and adders enables the GE-635 programmer to intermix efficiently single- and double-precision instructions.

The floating-point instruction repertoire includes two especially convenient divide instructions: Floating Divide Inverted (FDI) and Double-Precision Floating Divide Inverted (DFDI). These instructions cause the contents of the memory location to be divided by the contents of the AQ-Register--the reciprocal of other divide instructions in the repertoire. Thus, regardless of whether the contents of the AQ-Register must be a dividend or a divisor, the programmer can always perform a division without recourse to wasteful data movement operations.

Floating Negate, Normalize, Add to Exponent, and Single and Double Precision Compare Instructions further facilitate effective programming.

Transfer of Control

The complement of program transfer instructions permits the storage of the instruction counter in indexing registers X0 through X7 (at programmer option), an unconditional transfer, and conditional transfers. Conditional transfers on zero, plus, and carry have corollary transfers: non-zero, minus and no carry. Transfers on overflows and underflows are to maskable fault routines. If the normal fault routine is masked, transfer is at programmer option.

Special Operations

Several special instructions are provided to expand programmer options and to reduce coding work through utilization of hardware features.

Three repeat instructions in the repertoire provide unusual programming advantages: Repeat (RPT), Repeat Double (RPD), and Repeat Link (RPL). The RPT and RPD instructions permit execution of the next one or two instructions a selected number of times according to program requirements; they are especially useful for operating upon sequential lists in memory. For example, if RPT is used with any of several compare instructions to search a list, termination of the repeats will occur when a "hit" is made, according to programmer option as established by conditions in the RPT instruction. The "hit" causes transfer to the next sequential instruction.

Also, high-speed movement of large volumes of data can be accomplished by repeated execution of Load A/Q, Store A/Q, instructions, using the RPD command. Multiple executions of these instructions require memory accesses only for data to be moved.

The RPL instruction is similar in its execution to the RPT and RPD; it facilitates the processing of threaded lists scattered through memory.

The Effective Address to Register (EAR) instructions permit the effective address of such an instruction to be placed in any of the index registers, in the A-Register, or in the Q-Register. Thus, any effective address referenced frequently in a program can be stored in a register and

used without lost processing time in repeatedly redeveloping the effective address. Furthermore, the EAR instructions provide the GE-635 programmer with the ability to transfer data among any of the index registers, the A-Register, and the Q-Register.

The Binary to Binary Coded Decimal (BCD) instruction converts the magnitude of a 36-bit or smaller binary number to its decimal equivalent in BCD form. The conversion is made automatically, one decimal digit per instruction execution, using previously-stored conversion constants. The BCD form of the converted number is readily available for further operations.

The Gray to Binary (GTB) instruction converts a 36-bit word containing data in the Gray code (for example, coded analog information from an analog-to-digital input device) to its binary equivalent in only one execution of the instruction. This instruction enhances the use of the GE-635 in real-time applications.

The Execute Single and Execute Double (XEC and XED) instructions allow the programmer to execute remote instructions singly or in pairs. Because the address counter is not disturbed by the execute remote instructions, the program will continue sequentially after executing the XEC or XED referenced instructions, providing the referenced instructions do not alter the address counter. If a referenced instruction affects the address counter, a program transfer occurs.

Master Mode Operations

The Connect instruction is the only input/output instruction in the GE-635 repertoire. The Processor, having set up the I/O commands in the system memory, issues a Connect instruction to the Input/Output Controller which then assumes I/O responsibility.

All Master mode operations, including the Connect instruction, are reserved for the General Comprehensive Operating Supervisor and other special-purpose operations that are written either as extensions to GECOS or as a more limited replacement for GECOS.

IV. INPUT/OUTPUT CONTROLLER MODULES

In the GE-635 computation system, the Input/Output Controller (IOC) module is the coordinator of all input/output (I/O) operations between the complement of peripheral subsystems and up to four Memory modules. The Input/Output Controller operates essentially as a stored-program device controlled by and sharing memory with a Processor module. Data transfers between a peripheral device and memory are accomplished by the IOC while the Processor runs job programs. Peripheral devices are controlled by Processor-prepared instruction lists stored in the memory that controls the IOC.

Significant features of the Input/Output Controller include:

- Complete memory protection for all Input/Output Controller input/output data transfers.
- Total awareness of the number, types, and states of up to 16 I/O subsystems per Input/Output Controller.
- Hardware-software integration resulting from control programs designed to take full advantage of equipment features.
- Ability to interrupt Processor operations.
- Scatter-gather of randomly-stored I/O program data.
- A special I/O Processor for peripheral channel management.
- Maximum data transfer capacity of 1.6 million characters per second on up to 16 peripheral channels.
- Maximum single-channel data transfer capacity of 400,000 characters per second.
- Modular use of high-performance channels (up to 6) and standard-performance channels (up to 10).
- Simultaneous operation of up to 16 peripheral subsystems in a wide range of equipment configurations.
- Communication with up to four independent Memory modules.
- Channels available for data communications networks, such as the General Electric DATANET-30.
- The ability to handle data transfers for nonstandard I/O devices, without modification.
- A permanently-installed program for generating micro-commands from a Comprehensive Operating Supervisor macro-command; in this sense the Input/Output Controller possesses macro-processor features.

MACRO PROCESSOR FEATURES

The Input/Output Controller contains a Micro Program Generator that accepts macro-commands from the Comprehensive Operating Supervisor. The micro-commands from the generator control all activities of the Input/Output Controller, including operation of its I/O Processor.

The Micro Program Generator and I/O Processor constitute the essential equipment that enables Input/Output Controller operations to be independent of the Control Processor module. With this Input/Output Controller capability, I/O data transfers and media conversion operations can be carried on simultaneously with other object program Control Processor operations--true simultaneous multiple read-write-compute operations.

Micro Program Generator

The Micro Program Generator uses a permanently-installed program to produce control signals to all sections of the Input/Output Controller. The control signals are grouped into four basic routines: I/O initiation, I/O data servicing, I/O termination, and special external interrupts. To produce the sequence of control signals which make up these routines, macro-instructions from the Input/Output Supervisor in the Comprehensive Operating Supervisor in main memory are accessed by the Input/Output Controller, when needed, and transferred to the Micro Program Generator control subsection.

The Generator control provides input signals to a permanently-wired program matrix which, in turn, generates micro-command signals on 160 lines connected to all data-handling components of the Input/Output Controller and, in particular, to the I/O Processor.

Control lines from the I/O Processor carry the operation signals for:

1. Accomplishing the initiation, data servicing, and termination routines for operations involving a peripheral device
2. Moving data between main memory and Input/Output Controller and buffer storage
3. Sensing data transfer termination
4. Returning control to the Comprehensive Operating Supervisor.

I/O Processor

The I/O Processor contains decoding and control logic, a high-speed parallel adder, and auxiliary working registers for executing the micro-command instructions provided to it from the Micro Program Generator. The more important functions performed by the I/O Processor are:

1. Establishing connection between a Memory Controller and a specified peripheral device
2. Making limit tests to perform I/O memory protection
3. Regulating the transfer of characters for standard-performance peripheral operations and of words for high-performance peripheral operations between the Memory module and Input/Output Controller buffer storage

4. Making comparisons, upon data transfer terminations or exception conditions, that require micro-program jumps

5. Generating Input/Output Controller interrupt signals, sensed by the program interrupt cells in the Memory Controller, for eventual return to the Comprehensive Operating Supervisor.

PERIPHERAL EQUIPMENT CAPACITY

Equipment Classification

Peripheral subsystems connected to Input/Output Controllers are classified as high-performance or standard-performance, depending upon their data transfer rates. Up to six high-performance and ten standard-performance subsystems can be directed simultaneously by an Input/Output Controller, including I/O configurations with multiple consoles. Typical high-performance and standard-performance subsystems are listed below:

High-Performance Peripheral Subsystems	Standard-Performance Peripheral Subsystems
Disc Storage Unit	900-cpm Card Reader
Magnetic Drum Storage Unit	100-cpm and 300-cpm Card Punches
15,000/41,600-cps Magnetic Tape Units	1200-line-per-minute Printer
30,000/83,200-cps Magnetic Tape Units	Perforated Tape Reader and Punch
30,000/83,200/120,000-cps Magnetic Tape Units	Data Communication
	Console

Basic and Expanded Configurations

The basic complement of peripheral devices required to implement a GE-635 minimum on-line configuration with one Memory, one Processor, and one Input/Output Controller is serviced by three high-performance channels and five standard-performance channels (see Figure I-1). The high-performance channels service:

1. A Magnetic Drum or, alternately, a Magnetic Disc Storage Unit

2. A Dual-Channel Magnetic Tape subsystem with eight Magnetic Tape Units.

The standard-performance channels service:

1. A Card Reader

2. A Card Punch

3. Two Printers on separate channels

4. A central operations Console for communications between the operator and the Comprehensive Operating Supervisor.

This minimum hardware configuration, under complete control of the Comprehensive Operating Supervisor, externally stores (on magnetic disc or drum) and calls in all required system compilers and assemblers, stores and calls in object programs and requires no operator intervention other than handling input files and printer output.

This minimum equipment configuration can be expanded by the addition of peripheral devices that use up to six high-performance channels and up to ten standard-performance channels to accommodate the compiling, assembling, and running of additional object programs.

Simultaneous Channel Operation

The Input/Output Controller performs simultaneous I/O transfers on all available channels, as directed by the Comprehensive Operating Supervisor. The Supervisor continuously maintains in permanently protected memory up to 16 groups of channel control words, one group for each channel. These words act as macro-commands that are transferred to the Input/Output Controller to initiate I/O activities. After the macro-commands have been accepted and interpreted, the Input/Output Controller independently transfers data between the specified peripheral device and the job program associated with that device. Overlapping occurs so that I/O transfers, once initiated, continue simultaneously on all channels activated by the Comprehensive Operating Supervisor--up to a total of 16 channels or to the maximum Input/Output Controller throughput.

Data Communication Interface

Any of the standard-performance Input/Output Controller channels can be used as a data communication interface. The channel selected is known to the Comprehensive Operating Supervisor and is used with all job program I/O activities associated with the communications network controller. Outgoing data from the communications network job program is handled by the Supervisor and the Input/Output Controller in the same manner as I/O transfers to any other peripheral device connected through the Input/Output Controller. When the data communications controller has incoming data for the job program, the controller causes an automatic Processor interrupt via the Input/Output Controller. The interrupt causes the Comprehensive Operating Supervisor to initiate I/O data transfer from the controller to the associated job program.

DATA TRANSFER CAPACITY

The high total throughput capacity of the Input/Output Controller is achieved by high-rate servicing of the 16 channel terminals from all peripheral subsystems. A priority scheme is incorporated in the servicing so that, by selectable equipment interconnections, peripheral devices can be accommodated according to their transfer rates and to installation requirements.

The combined high-performance and standard-performance character transfer rate of the Input/Output Controller, operating at maximum capacity and allowing for time used in accessing data words and characters from memory, is approximately 1,600,000 characters per second. Under similar conditions, the maximum high-performance channel transfer rate is 400,000 characters per second; the standard-performance channel, 10,000 to 50,000 characters per second, depending upon the complement and configuration of peripherals used.

SCATTER-GATHER CAPABILITY

Associated with each input/output channel and stored in main memory may be scatter-gather lists (data control lists) that are part of each job program. The words making up these lists are referred to as data control words (DCW's) and have the format shown in Figure IV-1.

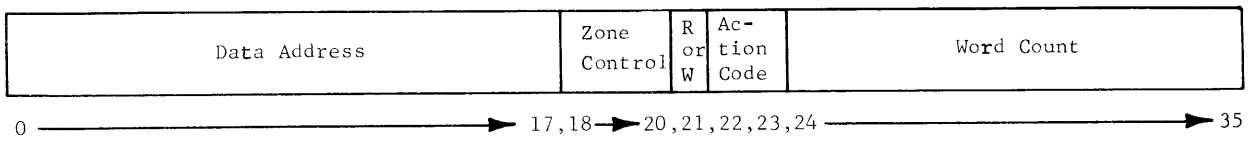


Figure IV-1. Data Control Word Format

The word count, data address, and action code fields contain job program information used by the Input/Output Controller. The word count is the number of words (per data block) to be transferred; it becomes the residual word count in the case of incomplete transfer of a data block. The data address is the location of the first word of a data block; it may be also the address of the first data control word of another data control list when the action code specifies a transfer to another list.

The action codes of a DCW and their functions are:

ACTION CODE	FUNCTION
00=Transmit and disconnect	The Input/Output Controller finishes data transfer and interrupts the Processor.
01=Transmit and proceed	The Input/Output Controller transfers current data and obtains the next DCW in the data control list.
10=DCW transfer	The Input/Output Controller obtains the DCW indicated in the data address field.
11=Non-Transmit and proceed	The Input/Output Controller bypasses the current data block and obtains the next DCW for processing the next data block.

The zone control and read/write (R or W) fields are managed by the Input/Output Controller during character transfer operations.

During write operations, data can be gathered from these blocks of memory locations and transferred to any peripheral device; analogously, data can be scattered during read operations. After completion of all data transfers associated with a scatter-gather list, control is returned via the program priority interrupt facility and Processor to the Comprehensive Operating Supervisor. Scatter-gather for magnetic tape writing operations on two tapes is shown in Figure IV-2.

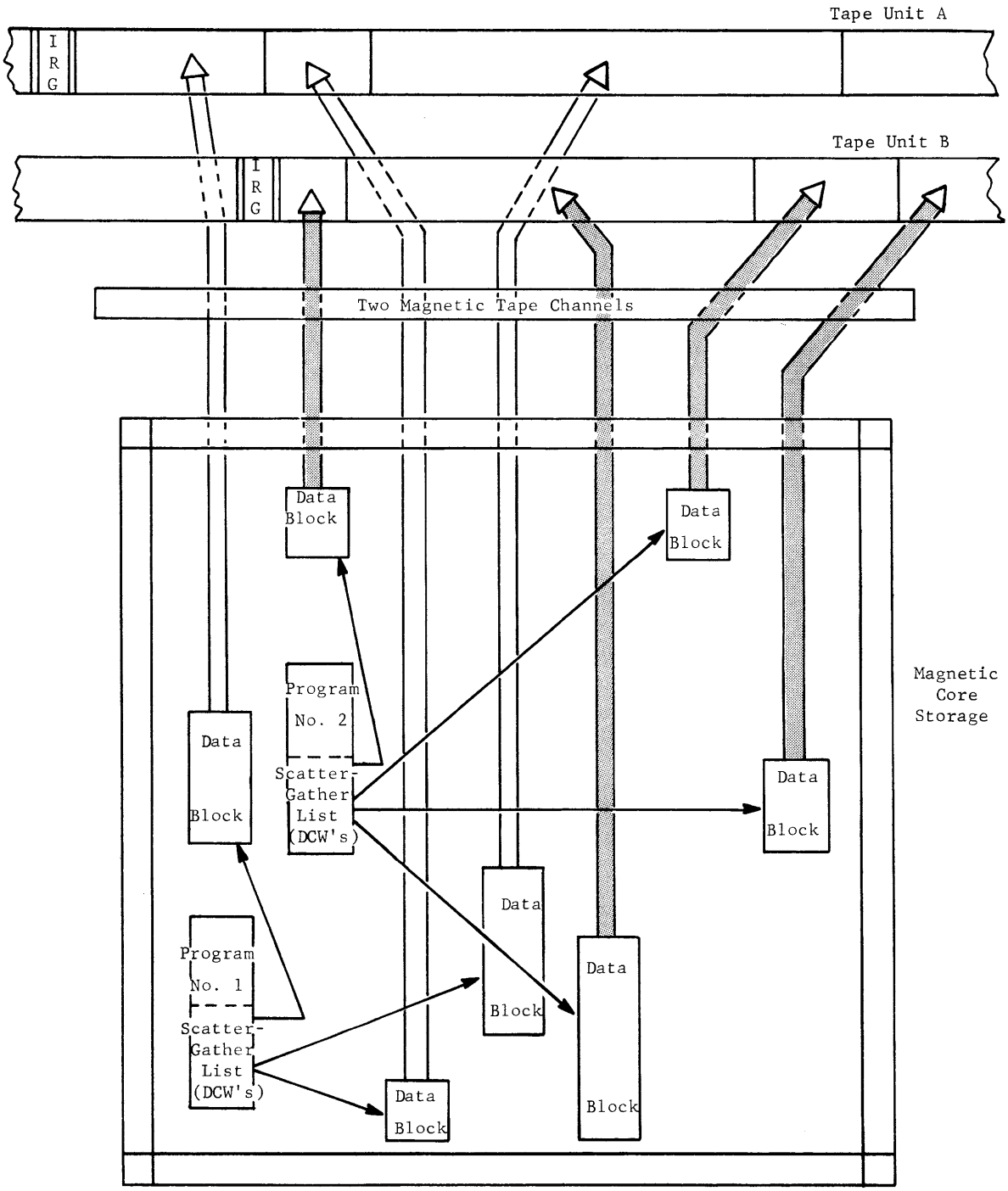


Figure IV-2. Gather Writing on Magnetic Tape

INPUT/OUTPUT CONTROLLER MEMORY PROTECTION

The data control list in each job program is composed of data control words that contain, among other data, information used by the Input/Output Controller to achieve I/O memory protection on all I/O transfers.

For each block of data in a job program there is a DCW containing the relative starting address and the word count of the block. In a normal data transfer, the Input/Output Controller adds the lower address limit (submitted from the Comprehensive Operating Supervisor) to the relative starting address to check the absolute starting address. The word count is added to the absolute starting address and compared with the upper address limit (also submitted from the Comprehensive Operating Supervisor). Thus, the lower and upper absolute address limits for each data block are checked; and memory protection is guaranteed. If any address limit is out of tolerance, the Input/Output Controller transfers to a fault routine.

INPUT/OUTPUT CONTROLLER PROGRAM INTERRUPTS

The Input/Output Controller can cause program interrupts of Processor operations under four conditions:

1. Failure to initiate an I/O activity
2. Termination of a channel busy status
3. Occurrence of special conditions (completion of magnetic tape rewind, development of ready status after manual attention to the printer, etc.)
4. Input/Output Controller inability to communicate with the Comprehensive Operating Supervisor without parity errors.

To accomplish a program interrupt related to an activity in any of these categories, an Input/Output Controller sends an execute interrupt signal to a related binary cell in the program interrupt register of the System Controller. The binary cell causes the System Controller logic to produce a unique signal which automatically puts the Control Processor in the Master mode and causes a program interrupt.

V. PROGRAMMING SYSTEM

The GE-635 programming system provided by the General Electric Company includes three classifications of programs:

- Operations Control--The General Comprehensive Operating Supervisor (GECOS) and associated routines, such as the Remote Terminal Supervisor, Input/Output Supervisor, File and Record Control, and Loader.

- Language Processors--The FORTRAN IV and COBOL-61 Extended compilers that provide input to a single MACRO Assembler. Subprograms written in FORTRAN, COBOL and MACRO Assembler language can be integrated at execution time into one object program.

- A complete library of mathematical and service routines as well as SIFT (FORTRAN II to IV translator) and PERT. Debugging routines which will permit selective dumping through control information are provided at either compile or load time.

GENERAL COMPREHENSIVE OPERATING SUPERVISOR

The basic philosophy underlying the development of the General Comprehensive Operating Supervisor (GECOS) is the desire to achieve a basic and workable system for the supervision of a multiprogramming environment. Extensions and modifications to GECOS by users are to be expected, and the modular design of the control programs and the accompanying documentation bear this in mind.

Characteristic of GECOS is the existence of a multiprogramming, uniprocessor environment, a full complement of peripheral types, multiple programs in memory simultaneously, remote terminal supervisory capacity through the DATANET-30, and the elimination of external input/output queues through a centralized input/output supervisor. GECOS modularity is conducive to expansion for encompassing advanced multiprocessing and mass random access storage techniques.

The General Comprehensive Operating Supervisor, by virtue of its functional modularity, can perform many of its operating functions simultaneously. This capability is essential to the multiprogramming environment. Programs are input sequentially to the system; as each module of GECOS completes its function, control is passed to the next appropriate module, freeing the preceding module for the next program. Thus, multiple programs may be simultaneously in process: one being input and examined for processing requirements, a second undergoing peripheral and memory allocation, a third conducting actual processing, a fourth having input/output performed, a fifth undergoing termination procedures, and a sixth having its final results output. At the same time, other programs can be queued in storage awaiting their turn with operating modules.

Scheduling and traffic control for action by the various modules of GECOS is performed automatically to assure optimum utilization of processor time, peripheral equipment, and memory.

GECOS is organized into five modules:

Input Media Conversion
Allocation
Monitor
Termination
Output Media Conversion

Each module, although functionally independent, interfaces with other modules. The more frequently used elements reside permanently in memory, while less frequently used sections are called from drum storage as temporary overlays, when needed.

As each job is presented to the system, it is processed first by the Input Media Conversion module which queues input jobs and constructs information files for subsequent use by the Allocation module. From Input Media Conversion, control is passed to the Allocation module for assignment of peripherals and memory. As each activity within a job has peripherals and memory assigned, and when it reaches the top of the queue, it is placed into execution. During execution, an activity interfaces with the Monitor module of GECOS which, as necessary, accesses other parts of GECOS to satisfy interface requirements. After an activity is executed, it is processed by the Termination module for any necessary postprocessing. Peripherals and memory that were assigned to that activity are de-allocated, and the activity leaves the system.

During activity execution, the system generates output on a standard system output file. The user may also place output on the system output file. System output file processing is the responsibility of the Output Media Conversion module.

Input Media Conversion

Input Media Conversion reads the job from the on-line card reader, interrogates all control card information, generates control tables to be used by the Allocation module, and records the job onto the magnetic drum. The control cards are uniquely identified by a $ symbol in the first column and a system control word in columns 8 through 13. Through the interpretation of the control cards, Input Media Conversion creates files on magnetic drum for the many activities within the job, and channels pertinent information into two tables for the use of the Allocation module.

When presented to the Input Media Conversion module, each job is considered to consist of one or more dependent activities. Allocation is based upon the individual activities of a job, and Input Media Conversion segregates these activities and presents them in an ordered manner to the Allocation module.

Allocation

The Allocation module of GECOS performs the scheduling and allocation of peripheral devices and memory for each individual job activity. The scheduling algorithm of the Allocation module provides a practical foundation for initiating multiprogramming in an installation. The algorithm is easy to understand and experience gained with its use can be easily applied, at any given

installation, to evolving modifications or extensions to meet local needs. Refinement of the scheduling algorithm by the user is practical because of the modular construction of GECOS.

The function of the Allocation module is governed by five guiding principles:

1. No peripheral device or storage is assigned to an activity until all the peripheral requirements for the activity can be satisfied; at that time, all peripherals are assigned at once.

2. No job activity is initiated until all preceding activities of that job are completed.

3. Look-ahead through job activities for peripheral allocation can proceed for n activities, where n is an adjustable parameter of the allocation.

4. An urgency criterion permits oft-delayed activities to get first consideration each time an allocation occurs.

5. The scheduling algorithm encourages an efficient blend of processing and input/output activities so that delays attributable to setup are minimized.

NORMAL ALLOCATION. The Allocation module utilizes two information files generated for each activity by the Input Media Conversion module. One file is the Allocator Table, which is a core table providing immediate reference information to the Allocation module. The second file is the Control Stack, which is a drum file containing complete peripheral requirements, operator instructions, core storage and running limits, and the activity definition.

As each activity is considered, the Allocation module scans the Allocator Table to determine the gross peripheral requirements for that activity. If the requirements are met, specific peripherals are allocated and instructions are issued to the operator from information contained in the Control Stack. This technique allocates peripherals to n activities in advance of execution, as determined by available peripherals, thus allowing operators to perform preparatory functions (such as mounting tape reels) while prior programs are in execution.

When the assigned peripherals are ready and sufficient memory is assigned to the activity, the Allocation module initializes the program and assigns the activity for in-process Monitor control.

URGENT-CASE ALLOCATION. The order of job allocation or scheduling is based upon a dynamic urgency concept. Initially, jobs are processed by the Allocation module as they are presented by the Input Media Conversion module; however, each time that a particular activity is scanned in the Allocator Table and its gross peripheral requirements cannot be met, the urgency of the job is incremented. When a threshold of urgency has been passed (that is, the urgency value has exceeded a predetermined limit), the entire job assumes maximum urgency and no other jobs receive consideration for allocation until the urgent job requirements are filled.

In summary, Allocation as described satisfies the first four principles listed at the beginning of this section. Regardless of the mix of processor-bound and input/output-bound programs to be processed, the Allocation module will attempt to assign components for complete utilization of the GE-635 system, thereby facilitating the achievement of effective multiprogramming.

Monitor

The Monitor oversees the execution of each activity. Its functions include:

- User communication with GECOS
- Processor-detected fault interpretation
- Control of Master mode entry from job programs
- Policing of the GECOS memory overlay area.

The user and operator interface with GECOS is performed under the Monitor module control. Operator/GECOS communication via the input/output typewriter is a function of Monitor.

The Monitor interprets the various fault conditions that can be detected by the Processor. In cases where the fault is of the type that can be recovered by the user program, the Monitor module allows user-provided routines to be substituted for GECOS fault routines. These routines can be activated by the user through a fault vector.

Master mode entries are also translated by the Monitor. The result of an entry translation causes the Monitor to call upon other GECOS modules and routines.

The least-used of the routines are not permanently stored in the GECOS reserved storage, but are contained in system drum storage and, as needed, brought into an overlay area of the GECOS memory area which is policed by the Monitor module.

Monitor also contains the system loader that is used to call system programs as required to process jobs presented to the system.

Termination

Individual activity and overall job termination is initiated by the Monitor module either at job program request or whenever it is determined that an activity must be removed because an error was detected by GECOS. A portion of the Termination module is an overlay that is brought in from drum storage and placed into the memory area reserved for the use of the Termination module. Termination provides these functions:

- Provides a postmortem dump if the termination was requested via a forced termination condition
- Communicates with the operator for the removal of files when needed
- Closes the systems output file
- Summarizes the output file information for Output Media Conversion
- Produces an accounting record on the systems output file
- De-allocates peripherals

- Deletes appropriate Input/Output Supervisor, Dispatcher, and Allocator Table entries.

When these functions have been accomplished, a transfer of control to that portion of Termination that resides in permanent GECOS storage occurs, where core storage is compacted. This is made possible through the use of the Base Address Register which makes all programs dynamically relocatable. After completing these tasks, the Termination module relinquishes control to the Allocation module for re-allocation of the released peripheral devices and memory.

Output Media Conversion

Two types of object program files are considered as normal for output media conversion: the systems output file containing multiple interspersed reports from one or more programs, and bulk output or large-volume reports stored as discrete files. The former are satisfactorily processed by the Output Media Conversion module of GECOS. The latter are more adequately handled by bulk media conversion routines called in by GECOS as needed.

The Output Media Conversion module functions to:

- Collect records on the systems output file for multiple programs
- Maintain record counts for each program
- Generate and record control information for output blocks
- Coordinate the utilization of output media and redirect output to alternate collectors, as needed.

As each program generates printer or punch output records, the records are recorded on the systems output file. When all output for a given program or report has been accumulated, the Output Media Conversion module, using the Input/Output Supervisor, reads the systems output file and performs the required output function. Control information recorded in the file will specify that output be to a standard peripheral device, such as a printer or card punch, or to a remote terminal device, utilizing the General Remote Terminal Supervisor.

GENERAL INPUT/OUTPUT SUPERVISOR

The General Input/Output Supervisor is the control program that services input/output requests for all programs using the Processor. The Monitor module, upon detection of a request for input/output service, accesses the Input/Output Supervisor and maintains proper time-sharing of components.

The General Input/Output Supervisor embodies several important system concepts. These are:

- Complete protection of peripheral files. One program cannot write or malposition files of another program.
- Standard detection and recovery actions are available on alert conditions, or conditions that require operator action.
- Automatic simulation of tape files on magnetic disc or drum, providing interchangeability between these media.

- Substitution of physical for symbolic addresses at execution time.
- Automatic measurement of peripheral time for accounting.
- Selection of an alternate program to be run in the event that an input/output request cannot be immediately satisfied.

Functions of the Input/Output Supervisor are performed by these sections:

Dispatcher
Roadblock
Courtesy Call
Forced Relinquish

Specific input/output actions needed are described by the job program, and the Input/Output Supervisor is entered to initiate each execution. The Input/Output Supervisor queues one command (or two, under certain circumstances) per peripheral device. If a program requests input or output on a device having a full queue, the Dispatcher portion of the Input/Output Supervisor gives control to another program that is not being delayed (or Roadblocked) by incomplete input or output. When such a change of control is accomplished, the Dispatcher stores the location of the Roadblocked request so that it can be reissued later. Whenever the Input/Output Supervisor performs an input or output operation or gives control to a program not previously in control, it increments the Processor time or peripheral time used by that program.

Roadblock

For programs that contain activities which cannot proceed until some input or output operation is completed, the user must make some provision for delay. The GECOS Roadblock entry automatically provides such a delay until all input and output requests initiated by a program are completed.

Only while a Processor is assigned to a program is processing time logged for that program for installation accounting. Only input/output time occuring after a program is Roadblocked is logged for that program.

From the machine installation viewpoint, the waiting period for peripheral action can be used profitably in advancing other programs. The resultant alternation of the Processor among several programs to keep it fully occupied doing useful work is a major advantage of multiprogramming. No additional programming burden is introduced to gain this advantage.

Dispatcher

Each time that a program yields control to GECOS because it is waiting for input/output to be completed, the Dispatcher has the opportunity to give control to another program in memory that can more effectively use the Processor.

For each program so considered, the following criteria are pertinent:

- How does the urgency of this program compare to that of other programs which qualify?
- Is this program Roadblocked?

- If input/output action was pending when the program relinquished control, has some input or output action been completed since then?

- Is the program actively seeking the use of the Processor?

In granting control to programs, the Dispatcher considers all GECOS programs to be more urgent than any job program. Programs are examined for a turn at the Processor in the order of their urgency.

Courtesy Call

The Courtesy Call enables a program needing much input/output action with only sporadic processing activity (such as file maintenance) to maintain just enough control to keep its peripherals operating efficiently. The programmer, when he initiates an input or output activity, asks that he be given control immediately after completion of that activity. In return, he obligates himself to yield the Processor to GECOS within 200 μsec (on the GE-635) or 400 μsec (on the GE-625). Should he fail to do so, his program is automatically terminated.

Forced Relinquish

An occasional Processor-bound program could compute for several minutes without ever needing to wait for input or output. It is desirable to interrupt this type of program periodically to allow input/output-bound programs a chance to initiate some input or output. For this reason, GECOS sets the timer on entry to a job program to interrupt whenever elapsed time exceeds 625 milliseconds, without relinquishing control. Control is then given to the Dispatcher, with the interrupted program being considered inactive until some other program has had its turn. The Forced Relinquish elapsed time is a variable parameter and may be redefined by the installation.

In summary, the Input/Output Supervisor, together with the capabilities of the Input/Output Controller module, perform those operations necessary to initiate peripheral actions, check for completion, analyze any difficulties that may occur, resolve conflicting requests, and supervise programs awaiting input or output action (Roadblocked programs).

GENERAL FILE RECORD CONTROL

The General File Record Control is another control program of the operations control category related to input and output operations. It is expected that most usage of peripherals will be accomplished by requests on General File Record Control which, in turn, operates using the Input/Output Supervisor.

File Record Control permits job programs to regard all input/output data as being composed of records and files. Within similar classes or devices, information can be requested in these terms entirely independent of recording medium. Suitable tables supplied to General File Record Control describe the location and format of each record and file. File Record Control interprets the request for a particular record and initiates a specific peripheral action to be transmitted to the Input/Output Supervisor for servicing.

All compilers and job programs generated by compilers access the input/output control programs through the General File Record Control, thus permitting device and format independence from the assignment of specific peripheral devices. Direct access to the Input/Output Supervisor is permitted in assembler language programs, but use of General File Record Control is recommended.

GENERAL LOADER

The General Loader is called upon whenever execution of object programs is requested. When loading is to be initiated, all available memory is allocated to the Loader. When sufficient memory is not available to contain the Loader and the program to be loaded, certain GECOS functions are temporarily discontinued and their memory areas are made available.

Multiple Subprogram Integration

The Loader accepts subprograms, relocates them into one contiguous program and establishes the required linkage to make them function as a complete program.

Object Time Debug

Debug tables are constructed from debug control cards prepared in symbolic form and loaded with programs. The tables enable specific conditions to be tested and, if met, to cause debug output data and dumps to be produced at selected locations within the program.

Overlays

Object program overlay segments may be stored during loading. During program execution, overlays are called by using appropriate calling sequences to GECOS.

Termination

When loading has been completed, the Loader terminates, releasing the unused portion of memory, and defining to GECOS the starting location of the program just loaded.

GENERAL REMOTE TERMINAL SUPERVISOR

The GERTS programming module supervises the reception of job programs from remote terminals, submits them to GECOS for processing, and returns the desired output. It also provides dynamic scheduling and interrupts of remote jobs.

Job programs are transmitted from remote terminals through the DATANET-30 to the GE-635. Jobs are accepted by GERTS and collected on magnetic disc or drum storage. When a complete job has been received, GERTS recomputes priorities for all waiting jobs it has received and the job currently being processed.

Assuming the job just received has a lower priority than the job being processed, when GECOS is ready to initiate processing of another job, GERTS submits the waiting job having the highest priority.

If a lower priority job is being processed when GERTS computes priorities, GERTS interrupts that job, returns it to storage for later processing, and initiates processing of the higher-priority job.

GECOS processes jobs from remote terminals just as it does local jobs. When the GECOS output control module encounters remote terminal outputs, it releases them to GERTS which in turn routes them through the DATANET-30 to the proper remote terminals.

MACRO ASSEMBLER

The GE-635 MACRO Assembler is a two-pass symbolic language assembler that accepts the output from both the GE-635 FORTRAN Compiler and the GE-635 COBOL Compiler. It also provides the programmer with the options of coding in the MACRO Assembler open-ended language or directly in machine-oriented symbolic instructions. Processing is oriented toward using drum storage and the machine-coded output is either on punched cards or magnetic tape. Editing and appropriate printouts are provided automatically.

The principal functions performed by the MACRO Assembler are:

- Assembly of all basic machine instructions for every user program, regardless of GE-635 configuration
- Translation of control and assembly-edit formatting pseudo-operations
- Recognition and translation of addresses that are absolute or relative to subprogram origin, to common storage, to labeled or block common storage, and to externally-defined symbols
- Production of relocatable or absolute binary subprograms that can be combined at load time
- Allowance for programmer-defined macro-instructions at assembly time
- Provision for accepting compressed symbolic decks plus any desired alter cards as input, and producing an updated compressed deck as output.
- Complete listing of assembled program, plus a symbol reference table, is provided as output.

COBOL

The GE-635 COBOL Compiler has been prepared following the specifications described in the Department of Defense manual, COBOL-61 Extended. In this specification, several parts of the language are termed "elective," while other parts are "required." The GE-635 COBOL Compiler implements all of the required, and much of the elective, language. Some of the more significant electives that are available to the user are:

- The REPORT WRITER and the SORT verb
- The arithmetic extension that allows for arithmetic and relational operators
- The corresponding option of the MOVE verb
- The COMPUTE, ENTER, and USE verbs.

The COBOL system encompasses two elements: the source program written in COBOL language, and the compiler that translates the source program into an object program for computer input. The source program can be written as one complete program or as a series of subprograms that can be bound together at load time. The compiler operates under GECOS control, producing an output for input to the MACRO Assembler which, in turn, produces a machine-language program.

The COBOL Compiler performs all input/output record and file processing through the General File Record Control. Input to the compiler is normally from magnetic disc or drum; output is to magnetic tape for subsequent output to a printer.

FORTRAN

The FORTRAN IV Compiler accepts the newest and most powerful features of FORTRAN.

The source program written in FORTRAN IV language is translated into assembly language by the compiler which adapts advanced computer techniques to the unique capabilities of the GE-635. The resulting object program is then converted by the MACRO Assembler into machine language. Emphasis has been placed on producing highly-efficient object programs, at the same time providing fast compilation and efficient debugging during object program development.

Some of the newest features of FORTRAN IV that are embodied in the GE-635 FORTRAN Compiler are:

- The ability to have variable-field input, utilizing the NAMELIST statement
- Variable-field output
- Compile DEBUG statements in the object program.

SORT/MERGE

The GE-635 Sort/Merge Program package is an efficient, generalized, versatile tool for applications involving files too large to be sorted in available core memory. The Sort/Merge Program accepts input from magnetic tape, disc, or drum. Normal output is on magnetic tape, with an alternative of output to a random access storage device. The program provides combined sorting and merging, or merging only. In the former, operation is in three phases: 1) initial sort, 2) initial merge, and 3) final merge. For merge-only operations, the third phase only is used, eliminating the need for a separate merge program.

The Sort/Merge Program performs algebraic sorts on numeric or alphanumeric keys of any length up to the length of the full record. Each field of the sort key can be specified to control the sort in either ascending or descending sequence.

INPUT/OUTPUT MEDIA CONVERSION

A Bulk Media Conversion Program is available on the system library and can be called using a standard set of control cards. The program can perform media conversion from any peripheral device to another. In interpreting the control cards, Bulk Media Conversion constructs the

appropriate file control blocks and buffers needed to initialize General File Record Control for performing the required conversion. Conversion capabilities include:

- Punched card to magnetic disc
- Punched card to magnetic tape
- Perforated tape to magnetic disc
- Magnetic tape to printer
- Magnetic disc to punched card
- Magnetic tape to punched card
- Magnetic disc to remote terminal
- Magnetic disc to magnetic tape
- Magnetic tape to remote terminal.

MATHEMATICAL ROUTINES

An important part of the GE-635 system library is the mathematical routines which include:

- Common function evaluation
- Matrix manipulation
- Curve fitting
- Polynomial root determination.

The commonly-used function evaluation routines include the normal and inverse trigonometric functions, exponentials, logarithms, and certain special functions. The extra precision available in floating-point operations enhances the usefulness of the mathematical routines in scientific and engineering applications.

SERVICE ROUTINES

The GE-635 software system includes an integrated set of service routines grouped into: 1) a magnetic tape maintenance module, 2) a magnetic disc or drum storage maintenance module, 3) debugging programs, 4) software maintenance, and 5) logical file and record maintenance.

Debugging capabilities of the system include:

- Post mortem dumping; a check-point dump in octal format is provided for general debugging.
- Dynamic dumping; at run time, the programmer has the option of taking "snapshot" dumps. These dumps are selective and all options of post mortem dumping are available. The programmer can turn off snapshots during execution.

- Debug capabilities; debug at load time is a function of the Loader which recognizes debug descriptive cards and generates and inserts code at the proper locations to output debug information. Debug at compile time is presently available only to FORTRAN. A debug request statement provides the programmer with easy access to a fixed-format output. A count specification and a simple conditional are contained in a statement that permits control of the debug output.

- Source language; update capabilities are provided for all compilers and assemblers. That is, it is possible to replace, add, or delete, source statements relative to a given source deck stored completely on magnetic disc, drum, or tape.

All software available with the GE-635 is maintained by the system editor maintenance package. The compilers, assemblers, GECOS, and other system packages are corrected through updating information and master files of source, object, and system files that are kept on magnetic tape. Initially, the magnetic drum is written from the system tape and is regenerated only when major changes to the system require updating the contents of the drum.

VI. REMOTE INPUT/OUTPUT AND REAL-TIME OPERATIONS

Data communication systems, essential to many existing and future computer applications, are readily integrated with the GE-635 system.

The General Remote Terminal (GERTS) programming module, working with GECOS, permits multiple remote terminals and computers at widely-separated locations to communicate with the GE-635. One-way or two-way communication over common and private carrier lines, coaxial cables, or via microwave transmission is channeled through a General Electric DATANET-30 real-time communication processor connected to a standard-performance Input/Output Controller channel.

Acceptable remote terminals include, but are not limited to:

- Single input/output devices (perforated tape readers/punches, card readers/punches, printers, Teletype keyboards, etc)
- General Electric DATANET terminals
- Other computer systems.

Virtually any digital transmitting or receiving device can be used as a GE-635 remote terminal.

The General Remote Terminal Supervisor takes full advantage of features within the GE-635 to enable the user (in various data communications applications) to achieve:

- Effective remote data processing--A multiprogramming environment can include a large number of users in both local and remote locations with diverse processing needs. GERTS, working in close relationship with GECOS, provides both local and remote users with dynamic job scheduling and job interrupts for remote object program processing.
- Radical reductions in turn-around time--Total turn-around time, the period between the submission of a job for processing and the receipt of the desired output, is a function of several factors:

  Time spent in awaiting processing
  Actual processing time
  Time spent in awaiting output
  Actual output time.

  Through the use of multiple local and remote input/output devices, simultaneous processing and input/output, and automatic job interrupts, the non-productive waiting time and output time can be reduced to a level commensurate with the high-speed processing capabilities of the GE-635 system.

DATANET-30 DATA COMMUNICATION PROCESSOR

The DATANET-30 is a real-time data communication processor that automatically receives and processes information from remote terminals for direct input into a computer complex, and that automatically transmits information to these terminals over common-carrier communication facilities. By taking full advantage of the latest developments in the transmission of digital data, the DATANET-30 can satisfy the data communication needs of the most stringent business, scientific and real-time applications.

DATANET-30 features include:

- Eighteen-bit word, binary operation mode
- Magnetic core memory with a 7 μsec cycle, available in 4k, 8k, or 16k word capacities
- Stored-program operation, with indirect addressing and multiple indexing
- Up to 128 input/output channels, with patch-plug adaptation to character and word length and to transmitting and receiving speeds.

The DATANET-30 can:

- Control data transmission over common-carrier facilities
- Control communication between remote terminals and the GE-635
- Conduct simultaneous transmit and receive operations with multiple remote terminals
- Operate with 5-, 6-, 7-, 8-, 14-, and 20-level codes
- Act as a message switching center between multiple remote terminals
- Perform automatic data accumulation and distribution.

REAL-TIME OPERATIONS

The GE-635 permits real-time and other non-standard devices to communicate with the system via the System Controller. The interface with the System Controller permits independent data movement and program initiation under control of the external device. Thus, external devices can operate in either the conventional processor-controlled mode or in a self-controlled mode that, for specialized applications, can optimize system performance.

To initiate data movements or program interrupts, the external device issues memory interrupt pulses. For data transfers, the device also specifies the memory locations involved. With external address generation, data transfers can proceed upon demand of the device without Processor intervention to establish a channel connection. Equally important in real-time applications is the use of address designation by the device to format data, allowing intermixed data to or from several sources to be located by source rather than by time of transmission.

When the device requests an execute interrupt, a specific program interrupt is initiated, safe-storing the old program and transferring to the new. By associating specific external events with interrupt control and corollary programs, the need for a control program can be minimized, particularly for high-frequency events.

Masking execute interrupts and holding their execution until a more convenient time makes possible the assignment of a priority to the programs associated with the execute interrupt control. Programs with higher priority mask all programs of lower priority upon entrance and remove the mask prior to exit.

Memory protection for real-time operations is accomplished with a file protect register that prohibits Processor access to protected blocks of storage that are either sequentially or randomly spaced in memory. The file protect register can be read or set only by GECOS via the Control Processor in the Master mode.

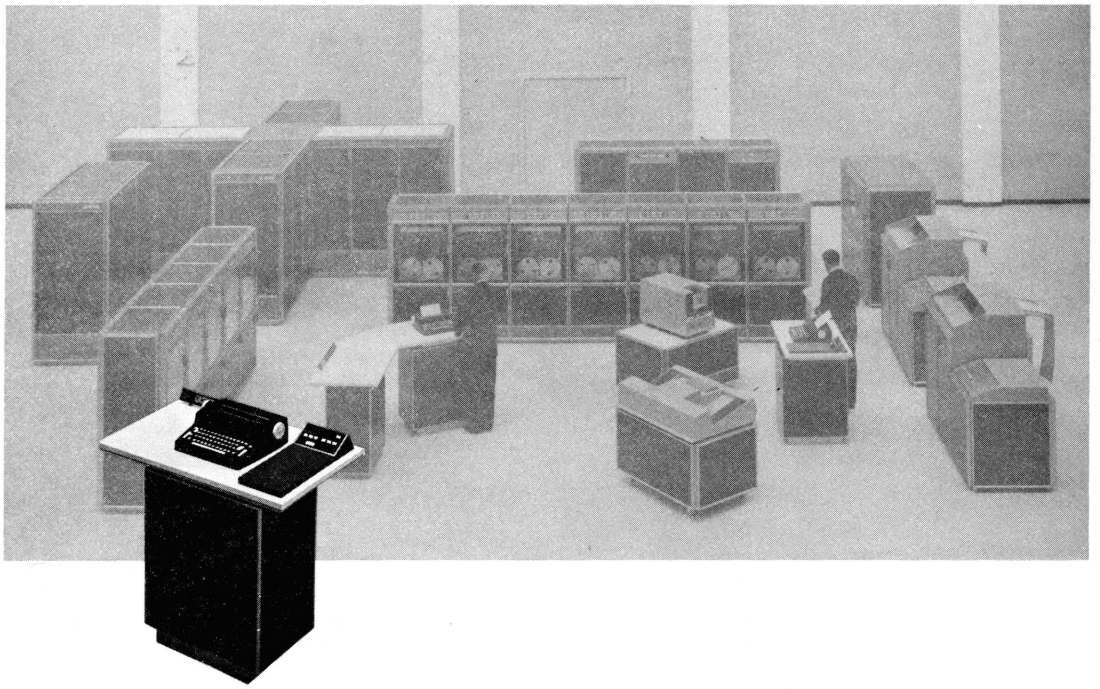
VII. PERIPHERAL EQUIPMENT SUBSYSTEMS

All peripheral subsystems (excluding real-time devices) communicate with the GE-635 system through an Input/Output Controller. Connection to the Input/Output Controller is through a standard interface that can accept a variety of peripheral device controllers, such as magnetic disc, drum, and tape controllers; card reader and punch controllers; and printer, perforated tape reader/punch, and data communication controllers.

The peripheral interface is designed as a universal interface for operation in a data processing environment. GECOS, through an Input/Output Controller, issues commands to the peripherals. Also, peripheral devices can request action by transmitting special interrupts to GECOS.

The peripheral interface is fundamentally character-oriented, transmitting 6-bit characters plus parity. High-performance peripheral devices communicate with memory through Input/Output Controller word buffers. A number of signal and status lines are also provided to permit the controlling system to operate and interrogate peripherals. The interface has a transmission rate capability of up to 1,000,000 characters per second.

CONTROL CONSOLES



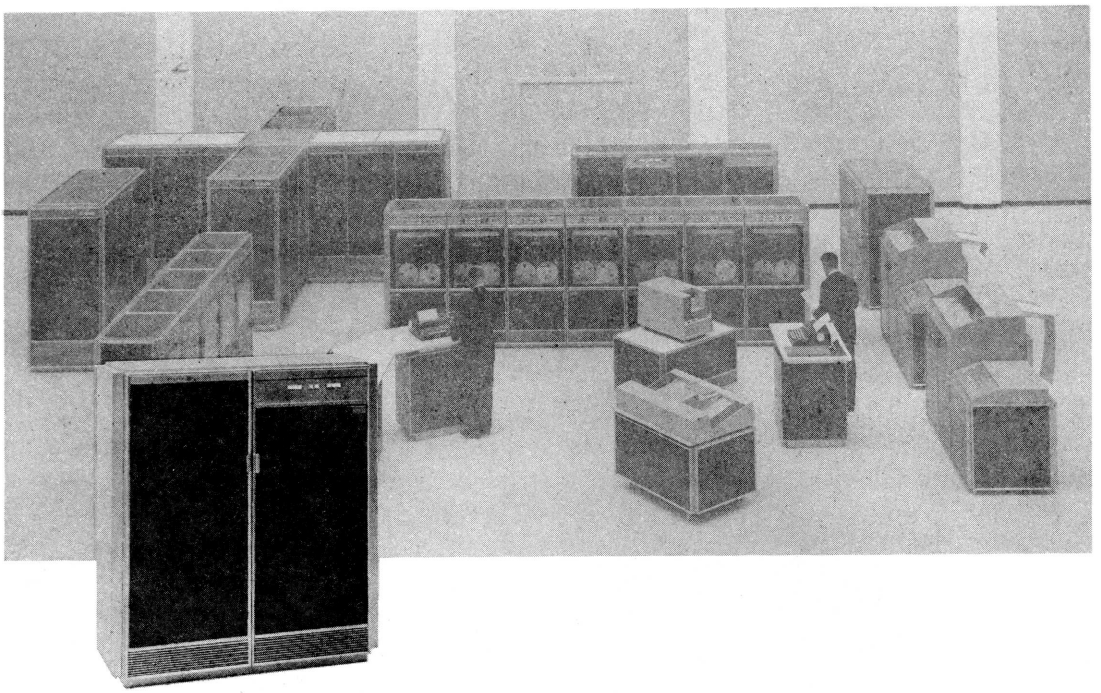
The primary function of a control console is to provide for direct communication between the operator and GECOS. The master console is a free-standing unit of suitable operator working height that connects to a standard-performance channel of the Input/Output Controller and that is controlled like a peripheral subsystem. Significant Processor and System Controller functions are displayed on the system status display panel, keeping the operator constantly informed of running status. The status display panel presents these lights and indicators:

SYSTEM READY light
OPERATOR ATTENTION light and switch
WAIT FOR INTERRUPT light
EXECUTED INSTRUCTIONS/SECOND indicator
CONSOLE READY light
MASTER MODE light
SLAVE MODE light
EMERGENCY POWER OFF switch

The console includes an input/output typewriter that accepts input data via keyboard entry and transmits the data to the Input/Output Controller. Output messages from the Input/Output Controller are printed at nominally 12 to 15 characters per second. An operating routine provided by GECOS provides responses to operator requests through the typewriter.

The standard console configuration is a single master console for operator-GECOS communication, and auxiliary consoles as required for subservient functions such as instructions to operators concerning peripheral conditions requiring attention. The auxiliary consoles are identical to the master console, except for the status display panel.

MD-20 MAGNETIC DRUM SUBSYSTEM

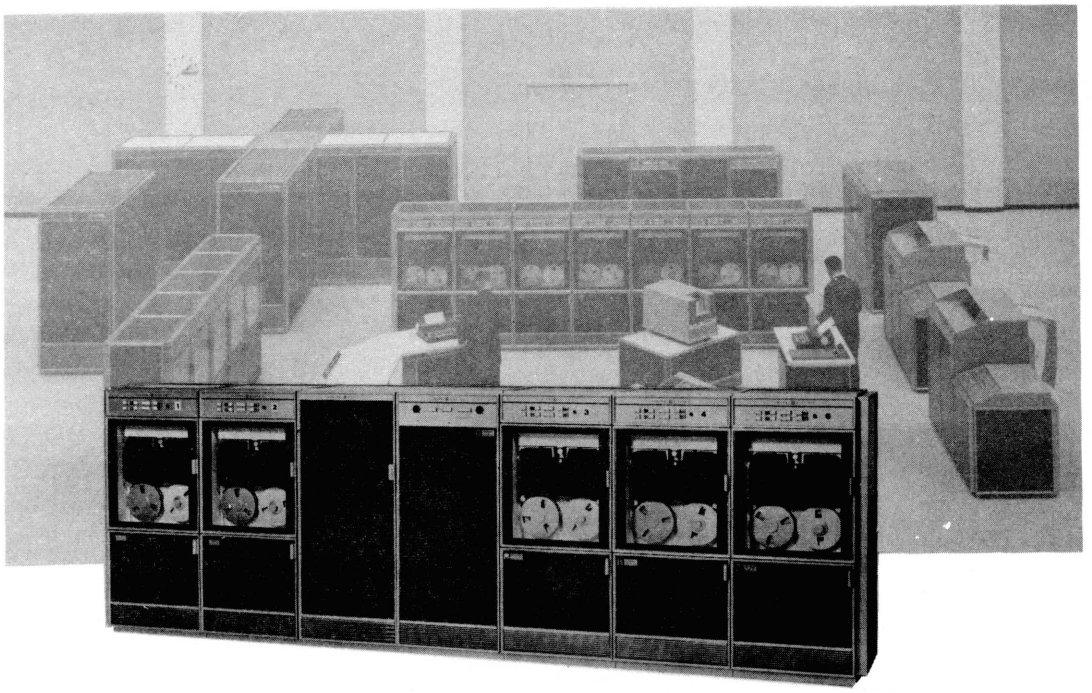


The MD-20 Magnetic Drum Subsystem is a high-performance auxiliary data storage facility. Communication with the GE-635 is through a high-performance channel on the Input/Output Controller at a rate of up to 370,000 characters per second. The storage capacity of one magnetic drum is six million characters (one million words). Average access time is 8.5 milliseconds. Data is stored in blocks of 384 six-bit characters, with all blocks addressed sequentially.

One peripheral interface channel is standard on the MD-20. An additional channel is available optionally. This additional channel is capable of independent command communication. The prime reason for two channels is the higher throughput made possible. When the second channel is used with another computer system, a protect system for data must be implemented in the control program.

The drum controller decodes each received command and initiates the proper sequence of command operations. The controller also monitors entire operations, including error checking, and transmits status information to the Input/Output Controller.

MAGNETIC TAPE SUBSYSTEMS



SUBSYSTEM CHARACTERISTICS

Subsystem	Maximum Number of Tape Units per Subsystem	Tape Motion (in ips)		Data Transfer Rates at Available Recording Densities (in bpi)		
		Forward	Rewind	200	556	800
MT-16	4 or 8	36	110	7 kc	20 kc	NA
MT-18	4 or 8	36	110	7 kc	20 kc	29 kc
MT-20	4 or 8	75	225	15 kc	42 kc	NA
MT-22	4 or 8	75	225	15 kc	42 kc	60 kc
MT-24	8 or 16	150	300	30 kc	83 kc	NA
MT-26	8 or 16	150	300	30 kc	83 kc	120 kc

DATA MEDIUM — Half-inch wide magnetic-oxide plastic tape, up to 2400 feet long

DATA FORMATS — Binary (standard) and special decimal

CHECKING

- Transfer timing
- Blank tape read
- Transmission parity
- Lateral parity
- Missing character
- Longitudinal parity
- Bit detected during erase

FEATURES

Either single-channel or dual-channel control of tape units.

Either two-density (MT-16, MT-20, and MT-24) or three-density (MT-18, MT-22, and MT-26) tape units.

Program or operator control of recording density.

Special decimal mode provides compatibility with non-General Electric tape systems.

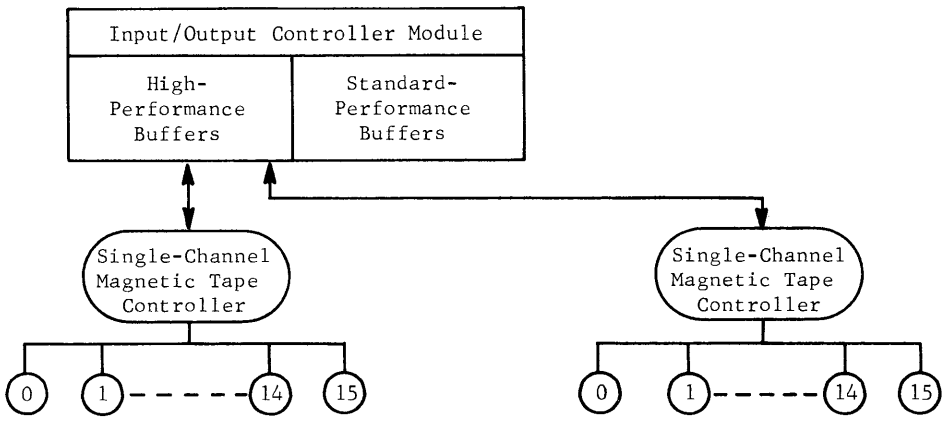
Dual-gap read-write heads for read-after-write checking.

Controller buffering during data transfers.

File protection through use of a write-permit ring.

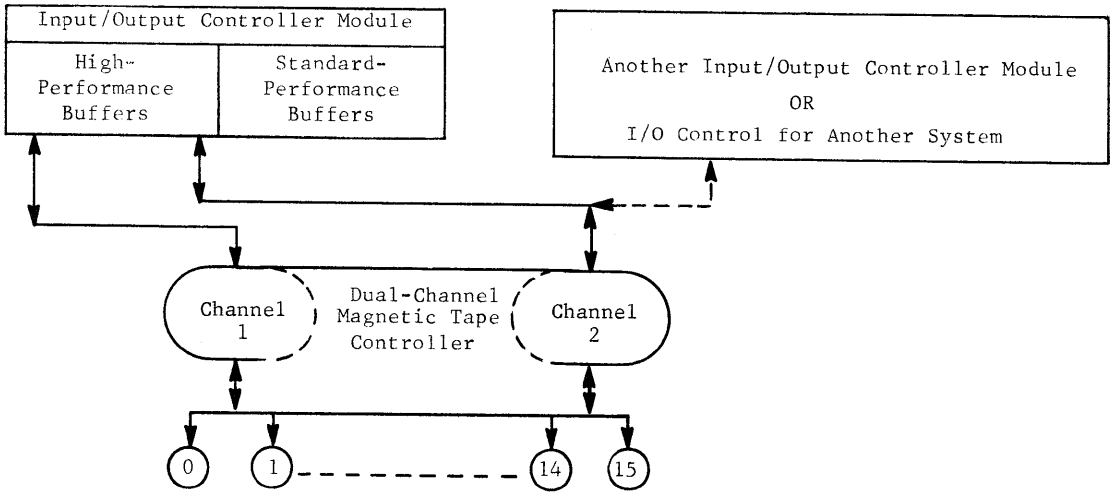
MAGNETIC TAPE SUBSYSTEM CONFIGURATIONS

Single-Channel Subsystems



A single-channel magnetic tape subsystem connected to a high-performance channel of an Input/Output Controller permits reading or writing on any one of up to 16 magnetic tape units connected to that controller. Reading or writing proceeds simultaneously with other peripheral operations on other peripheral channels and with Processor operations. Multiple single-channel tape subsystems make possible simultaneous read-read, read-write, or write-write operations.

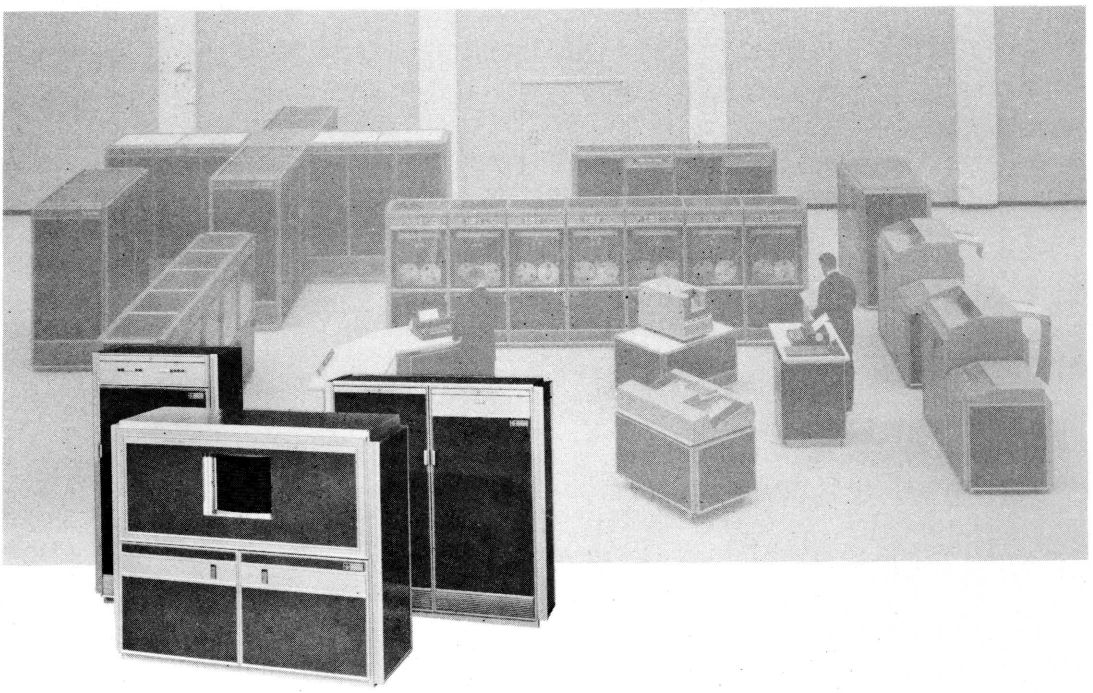
Dual-Channel Subsystems



Dual-channel control of a magnetic tape subsystem gives the programmer flexibility in selection of tape unit configurations for simultaneous or sequential read-write operations without consideration for tape unit channel assignments. As shown above, a subsystem including a dual-channel controller and two or more tape units permits accessing any tape unit through either channel and tape controller. Thus, if one controller is busy with an assigned tape unit, access can be gained to any other tape unit on the subsystem through the other controller. Switching of tape unit control between controllers can be automatic, reducing tape handling errors and operator time.

As shown by the dotted line, a magnetic tape subsystem featuring dual-channel control can be connected between two separate 600-series or other compatible systems through separate input/output channels. Either system can gain access to all tape units connected to the dual-channel controller.

DS-20 DISC STORAGE SUBSYSTEM



DATA MEDIUM AND FORMAT

Each disc storage unit (DSU) contains 16 discs, providing 32 recording surfaces. Each surface has two zones with 128 tracks each. Inner tracks can contain 320 words each; outer tracks, 640 words each. Thus, each DSC has a 23,592,960 character capacity.

SPEEDS

Data transfer rates (ave), 41,700 cps inner zone; 83,400 cps outer zone.

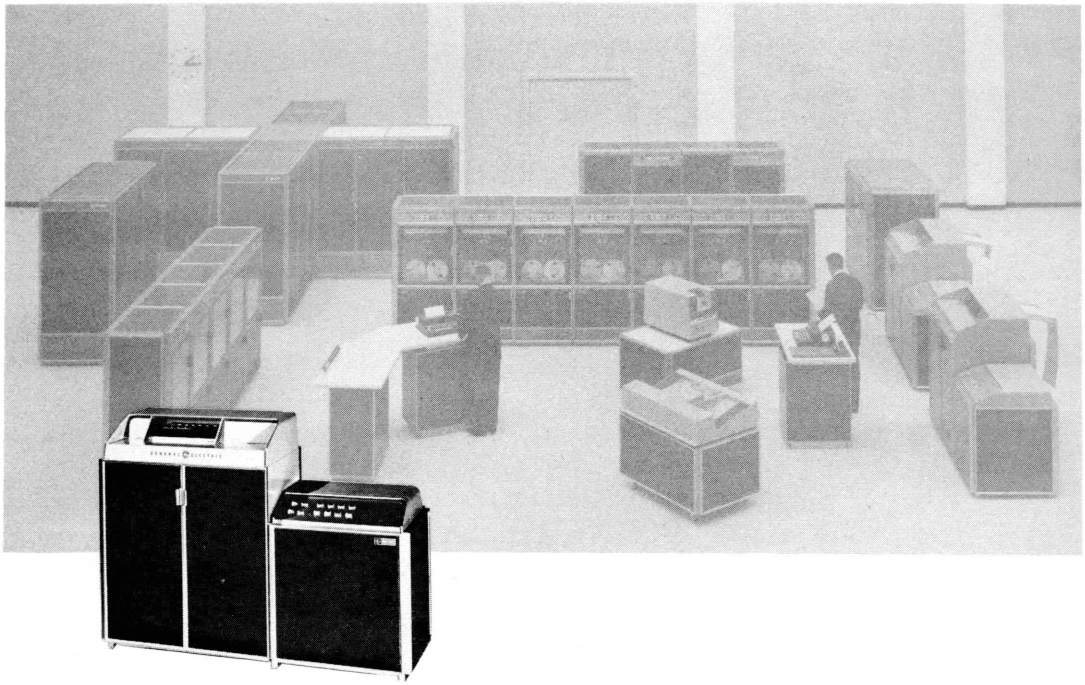
File latency time (one revolution) 51.3 ms.

Access time, block specified: 225 ms, average; 357 ms, maximum; block on adjacent track, 150 ms, maximum.

CHECKING

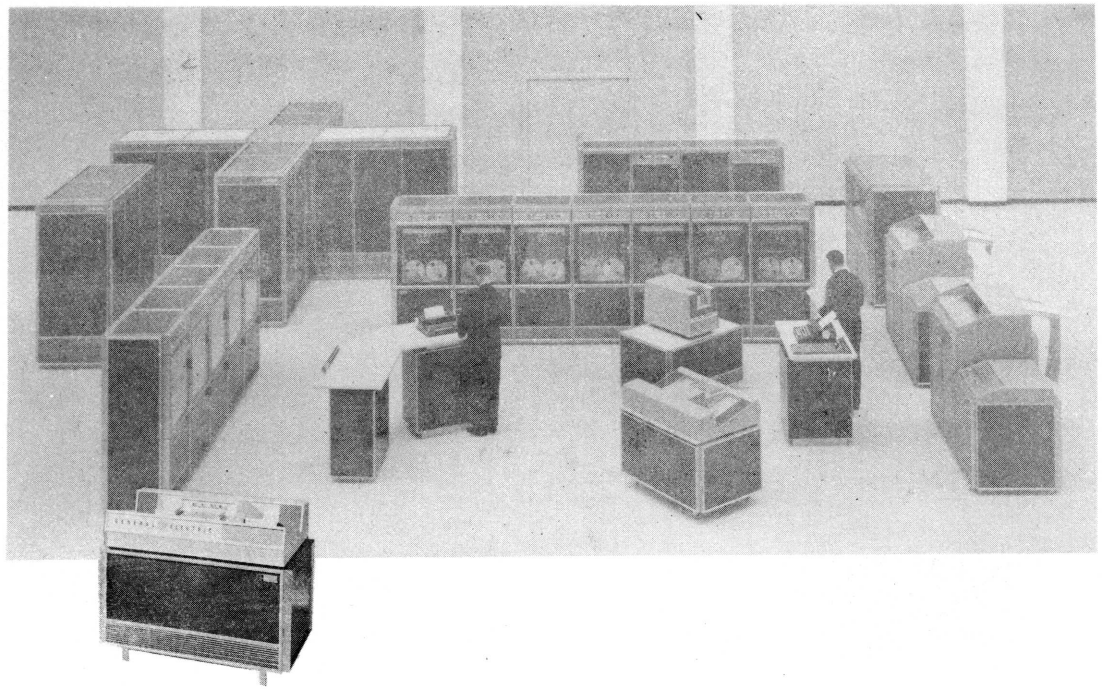
Transfer timing
Invalid control character
Check character
Buffer section committed
Invalid device code
Transmission parity
Internal error
Sector compare failure
Invalid operation code
Illegal buffer address

PR-20 PRINTER SUBSYSTEM



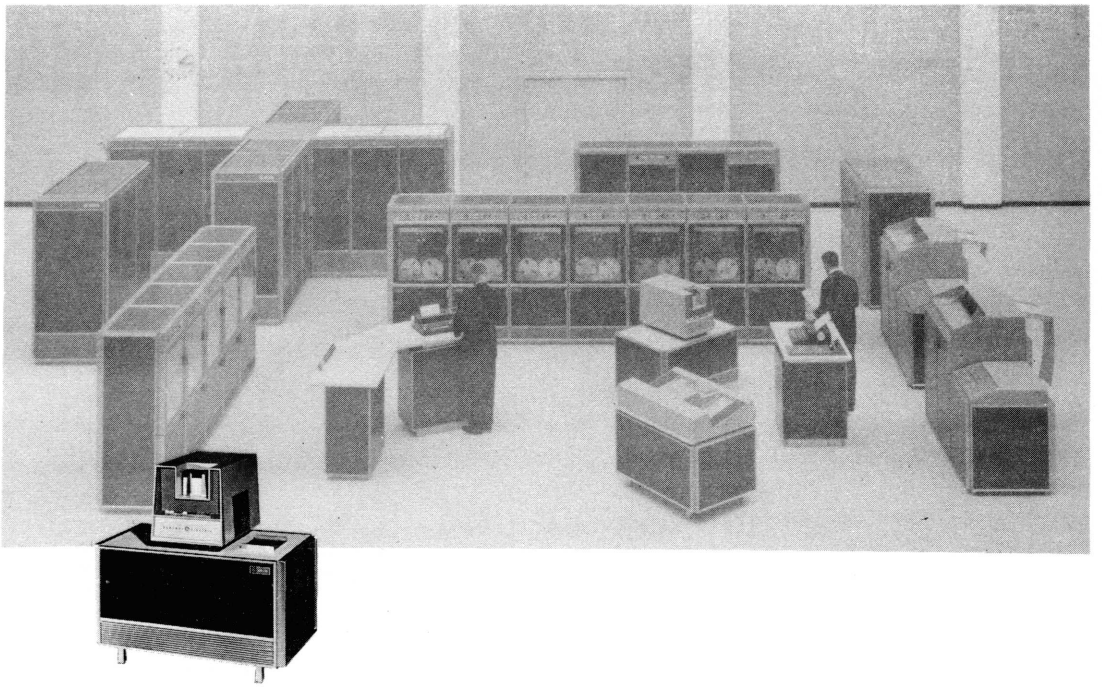
DATA MEDIUM	Continuous forms, 3 to 19 inches wide, up to 22 inches long, 1 to 5 copies (original and 4 carbons)
SPEEDS	1200 lpm, printing 46 most-used (adjacent) characters; 938 lpm, printing all 64 standard characters.
DATA FORMATS	Points open gothic font, 10 characters per horizontal inch, 136 characters per line; operator selection of 6 or 8 lines per vertical inch
OPERATIONAL MODES	Edit mode deletes edit symbols from print line; programmed blank insertion and slewing. Non-Edit Mode prints special characters in print line data for debug memory dumps.
CHECKING	Parity on data characters and on Vertical Format Unit (VFU) tape punch configurations.
FEATURES	Photoelectric sensing of the Vertical Format Unit (VFU) Tape for reliability
	VFU prevents runaway slewing
	Separate VFU mechanism for each mode of vertical line density
	Programmed-control of slewing by VFU Tape or countdown, includes top of page slew

CR-20 CARD READER SUBSYSTEM



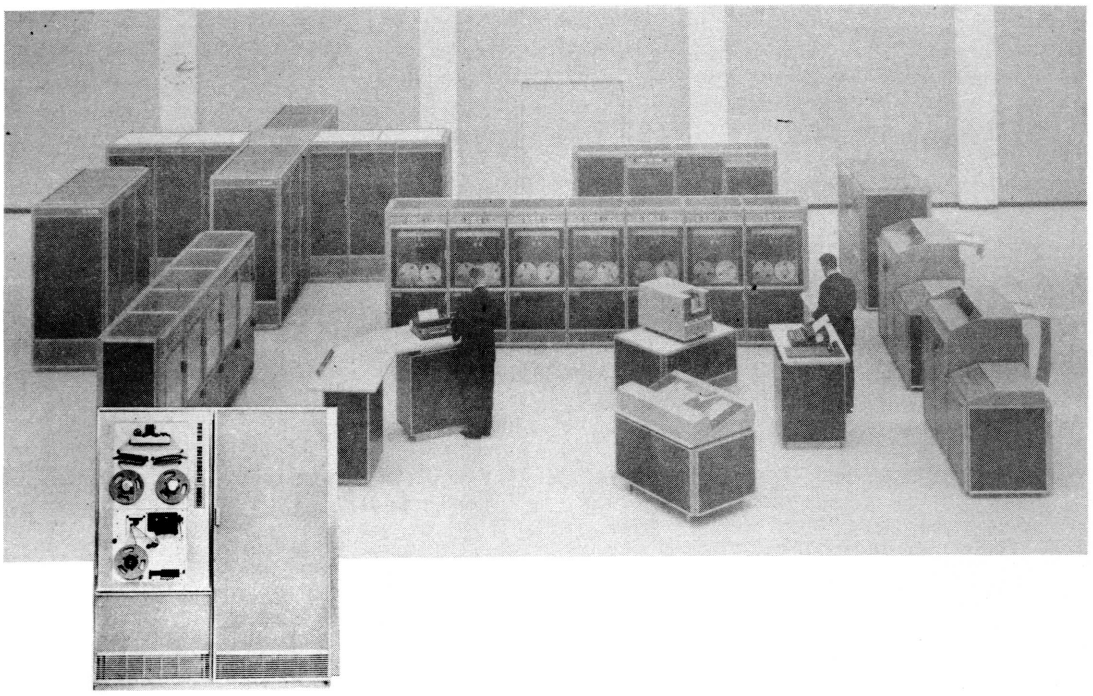
DATA MEDIUM	Standard 80-column cards with round or square corners	
SPEED	900 cards per minute under program-controlled feed	
DATA FORMATS	Standard Hollerith card code and column binary	
OPERATIONAL MODES	On-line decimal (Hollerith)	
	Binary (memory image)	
	Decimal and binary intermixed	
CHECKING	Card feed	Character validity (decimal mode)
	Card synchronization	Hopper empty
	Read head alert	Stacker full
	Dual-read and compare	Card jam
FEATURES	2000-card hopper and stacker capacity	
	Continued operation while loading or removing cards	
	Card reading simultaneous with other peripheral and Processor operations	
	Extensive error monitoring for high-accuracy data transfers	
	Reads intermixed Hollerith and binary cards, special character controlled	
	Last batch control via LAST BATCH switch	

CP-10/CP-20 CARD PUNCH SUBSYSTEMS



DATA MEDIUM	Standard 80-column with round or square corners
SPEED	CP-10, 100 cpm; CP-20, 300 cpm
DATA FORMATS	Standard Hollerith card code, edited Hollerith, and 12-row binary
OPERATIONAL MODE	On-line decimal (Hollerith) or binary
CHECKING	Card feed Card synchronization Parity Card jam Hopper empty Stacker full Chad box absent Chad box full Read-after-punch
FEATURES	CP-10, 800-card stacker and hopper capacity; CP-20, 3500-card stacker and 3000-card hopper capacities. Continued operation while loading or removing cards Card punching simultaneous with other peripheral and Processor operations Extensive error monitoring for high-accuracy data transfers

PT-20 PERFORATED TAPE SUBSYSTEM



DATA MEDIUM	Paper or Mylar* bonded tape, perforated with chad-type holes
SPEEDS	Reader, 500 characters per second; punch, 150 characters per second.
DATA FORMATS	Reads and punches 5-, 6-, 7- and 8-channel tapes, 10 characters per inch, in widths of 11/16, 7/8, and 1 inch
	Recognizes all possible code combinations
OPERATIONAL MODES	On-line: punching fully controlled by the Input/Output Controller (reading controlled by the Input/Output Controller and plugboard)
	Off-line: reading and punching controlled by the TS-20 Tape Reader/ Punch control panel
CHECKING	Optional odd or even parity while reading Odd parity check on all characters punched Input spool low Output spool full Tape breakage
FEATURES	Removable plugboard permits optional selection of codes, sprocket hole positions, and operational controls
	Extensive error monitoring for high-accuracy data transfers
	Reading or punching simultaneous with other peripheral and central processor operations
	Photoelectric reading mechanism for accurate, reliable reading without tape wear

*Trademark of E. I. duPont de Nemours & Co., Inc.

APPENDIX

INSTRUCTION SUMMARY

Instruction Classes

The GE-635 instruction repertoire is presented in groups as follows:

Data movement
Arithmetic
Shifting
Logic
Compare
Control
Special
External control

Data movement (floating-point)
Arithmetic (floating-point)
Control (floating-point)

Notes on Execution Times

Unless otherwise indicated, normal execution times include instruction accessing and execution.

Address modification effects on execution times are:

MODIFICATION TYPE	ADDITIONAL EXECUTION TIME
Register	0 μs for X0-X7, A, Q, IC, N
Register then Indirect	-0.5 μs for DU and DL* 1.1 μs per indirect cycle
Indirect then Register	1.7 μs per indirect cycle
Indirect then Tally	2.5 per indirect cycle with restore of indirect word

*No saving is effected when overlapping occurs. See 6 on the following page.

Instruction times are based on accessing instructions from memory in pairs, with overlapping of fetching, decoding, modification, and execution of the two instructions. Certain operations prevent overlapping; accordingly, the following timing adjustments should be made.

1. If an instruction in an even location alters an indexing register used as an address modifier by the next instruction, 0.8 microseconds will be added to the second instruction.

2. If an instruction in an even location alters the next instruction (for example, replaces the next instruction with a new instruction in memory), 1.7 microseconds will be added to next instruction.

3. If a transfer instruction is in an odd location, 0.5 microseconds will be added to the instruction.

4. If the first instruction following a program transfer instruction is in an odd location, 0.8 microseconds will be added to this first instruction.

5. If either a transfer type instruction in an even/odd location or a store type instruction in an odd location is preceded by a sequence of overlapping operations (types like fixed-point divisions, multiplications, shifts, etc., and all floating-point operations except "loads"), 1.0 to 2.0 microseconds will be added to the transfer or store instruction depending on the types of instructions in the program coding sequence preceding the instruction.

6. If an instruction specifying a DU or DL modification variation is preceded by overlapping operations (see item 5 above), the normal saving of -0.5 microseconds will be sacrificed.

7. If a "store" type instruction (see item 5 above) is in an even location, 0.5 microseconds will be subtracted from the listed normal time; if it is in an odd location, 0.5 microseconds will be added (except for the condition given in item 5 above).

8. If a "store" instruction is followed by one or more "store" instructions, 0.5 microseconds will be subtracted from each such following instruction.

9. Floating-point alignment and/or normalization operations and shift operations times are average times based on a mean number of shift cycles--five for both types of operations--required to cover the number range -2^n to 2^n-1. Actual times may vary 0.8 microseconds at approximately 0.2 microseconds per shift cycle.

10. Where unnormalized numbers are used in normalized floating-point operations, worst-case conditions requiring a maximum of seven normalizing shifts can add as much as an additional 1.5 microseconds to the operation.

In items 3 through 7 above the penalty can be avoided by locating the instructions at even addresses. Alternatively, for overall program timing estimates, one-half the penalty can be used on the assumption that the instruction will fall in an odd location 50 percent of the time.

Mnemonic Code		Normal Execution Time (μsec) GE-625	GE-635
DATA MOVEMENT INSTRUCTIONS			
LDA	Load A	3.0	1.8
LDQ	Load Q	3.0	1.8
LDXn	Load Xn	3.0	1.8
LDAQ	Load AQ	3.0	1.9
LCA	Load Complement A	3.0	1.8
LCQ	Load Complement Q	3.0	1.8
LCAQ	Load Complement AQ	3.0	1.9
LCX_n	Load Complement Xn	3.0	1.8
LDI	Load Indicator Register	3.0	1.8
LBAR[1]	Load Base Address Register	3.0	1.8
STA	Store A	3.5	2.5
STQ	Store Q	3.5	2.5
STXn	Store Xn	3.5	2.5
STAQ	Store AQ	3.5	3.0
STC1	Store Instruction Counter Plus 1	3.5	2.9
STC2	Store Instruction Counter Plus 2	3.5	2.9
STZ	Store Zero	3.5	2.5
STI	Store Indicator Register	3.5	2.9
SBAR	Store Base Address Register	3.5	2.9
STCA	Store Characters of A	3.5	2.5
STCQ	Store Characters of Q	3.5	2.5
ARITHMETIC INSTRUCTIONS			
ADA	Add to A	3.0	1.8
ADQ	Add to Q	3.0	1.8
ADAQ	Add to AQ	3.0	1.9
ADXn	Add to Xn	3.0	1.8
ADLA	Add Logic to A	3.0	1.8
ADLQ	Add Logic to Q	3.0	1.8
ADLAQ	Add Logic to AQ	3.0	1.9
ADLXn	Add Logic to Xn	3.0	1.8
ADL	Add Low to AQ	3.0	1.8
ASA	Add Stored to A	4.0	2.8
ASQ	Add Stored to Q	4.0	2.8
ASXn	Add Stored to Xn	4.0	2.8
AWCA	Add with Carry to A	3.0	1.8
AWCQ	Add with Carry to Q	3.0	1.8
AOS	Add One to Storage	4.0	2.8
SBA	Subtract from A	3.0	1.8
SBQ	Subtract from Q	3.0	1.8
SBAQ	Subtract from AQ	3.0	1.9
SBXn	Subtract from Xn	3.0	1.8
SBLA	Subtract Logic from A	3.0	1.8
SBLQ	Subtract Logic from Q	3.0	1.8
SBLAQ	Subtract Logic from AQ	3.0	1.9
SBLXn	Subtract Logic from Xn	3.0	1.8
SSA	Subtract Stored from A	4.0	2.8

[1] Functions as a NOP in the Slave mode

Mnemonic Code		Normal Execution Time (μsec) GE-625	GE-635
ARITHMETIC INSTRUCTIONS (continued)			
SSQ	Subtract Stored from Q	4.0	2.8
SSXn	Subtract Stored from Xn	4.0	2.8
SWCA	Subtract with Carry from A	3.0	1.8
SWCQ	Subtract with Carry from Q	3.0	1.8
NEG	Negate	2.0	1.3
NEGL	Negate Long	2.0	1.3
MPY	Multiply Integer	7.0	7.0[1]
MPF	Multiply Fraction	7.0	7.0[1]
DIV	Divide Integer	14.5	14.2[1]
DVF	Divide Fraction	14.5	14.2[1]
SHIFTING INSTRUCTIONS			
ALS	A Left Shift	2.0	1.8
ARS	A Right Shift	2.0	1.8
ARL	A Right Logic	2.0	1.8
QLS	Q Left Shift	2.0	1.8
QRL	Q Right Logic	2.0	1.8
QRS	Q Right Shift	2.0	1.8
LLS	Long Left Shift	2.0	1.8
LRL	Long Right Logic	2.0	1.8
LRS	Long Right Shift	2.0	1.8
ALR	A Left Rotate	2.0	1.8
QLR	Q Left Rotate	2.0	1.8
LLR	Long Left Rotate	2.0	1.8
LOGIC INSTRUCTIONS			
ANA	AND to A	3.0	1.8
ANQ	AND to Q	3.0	1.8
ANAQ	AND to AQ	3.0	1.9
ANXn	AND to Xn	3.0	1.8
ANSA	AND to Storage A	4.0	2.8
ANSQ	AND to Storage Q	4.0	2.8
ANSXn	AND to Storage Xn	4.0	2 8
ORA	OR to A	3.0	1.8
ORQ	OR to Q	3.0	1.8
ORAQ	OR to AQ	3.0	1.9
ORXn	OR to Xn	3.0	1.8
ORSA	OR to Storage A	4.0	2.8
ORSQ	OR to Storage Q	4.0	2.8
ORSXn	OR to Storage Xn	4.0	2.8
ERA	Exclusive OR to A	3.0	1.8
ERQ	Exclusive OR to Q	3.0	1.8
ERAQ	Exclusive OR to AQ	3.0	1.9
ERXn	Exclusive OR to Xn	3.0	1.8
ERSA	Exclusive OR to Storage A	4.0	2.8

[1] Operations Unit Execution Time Only

Mnemonic Code		Normal Execution Time (μsec) GE-625	GE-635
LOGIC INSTRUCTION (continued)			
ERSQ	Exclusive OR to Storage Q	4.0	2.8
ERSXn	Exclusive OR to Storage Xn	4.0	2.8
COMPARE INSTRUCTIONS			
CMG	Compare Magnitude	3.0	1.8
CMPA	Compare with A	3.0	1.8
CMPQ	Compare with Q	3.0	1.8
CMPAQ	Compare with AQ	3.0	1.9
CMPXn	Compare with Xn	3.0	1.8
CANA	Comparative AND with A	3.0	1.8
CANQ	Comparative AND with Q	3.0	1.8
CANAQ	Comparative AND with AQ	3.0	1.9
CANXn	Comparative AND with Xn	3.0	1.8
CNAA	Comparative Not AND with A	3.0	1.8
CNAQ	Comparative Not AND with Q	3.0	1.8
CNAAQ	Comparative Not AND with AQ	3.0	1.9
CNAXn	Comparative Not AND with Xn	3.0	1.8
CMK	Compare Masked	3.0	2.2
CWL	Compare with Limits	3.0	2.2
CONTROL INSTRUCTIONS			
EAA	Effective Address A	2.0	1.3
EAQ	Effective Address Q	2.0	1.3
EAXn	Effective Address to Xn	2.0	1.3
RET	Return	4.0	3.3
TSXn	Transfer and Set Xn	3.0	1.8
TSS	Transfer and Set Slave Mode	2.0	1.7
MME	Master Mode Entry	3.0	2.3
DRL	Derail	3.0	2.3
TRA	Transfer Unconditionally	2.0	1.7
TOV	Transfer on Overflow	2.0	1.7
TQO	Transfer on Quotient Overflow	2.0	1.7
TZE	Transfer on Zero	2.0	1.7
TNZ	Transfer on Not-Zero	2.0	1.7
TMI	Transfer on Minus	2.0	1.7
TPL	Transfer on Plus	2.0	1.7
TRC	Transfer on Carry	2.0	1.7
TNC	Transfer on No Carry	2.0	1.7
TTF	Transfer on Tally Run-out Indicator OFF	2.0	1.7
TEO	Transfer on Exponent Overflow	2.0	1.7
TEU	Transfer on Exponent Underflow	2.0	1.7
TSZ	Transfer and Store in Zero	3.0	1.8
XEC	Execute	2.0	1.7
XED	Execute Double	2.0	1.7
NOP	No Operation	2.0	1.1
DIS	Delay until Interrupt Signal	2.0	1.7
SZN	Set Zero and Negative Indicator from Memory	3.0	1.8

Mnemonic Code		Normal Execution Time (μsec) GE-625	GE-635
SPECIAL INSTRUCTIONS			
BCD	Binary to Binary-Coded-Decimal	4.0	3.4
GTB	Gray to Binary	9.0	8.5[1]
RPT	Repeat	2.0	1.3
RPD	Repeat Double	2.0	1.3
RPL	Repeat Link	2.0	1.3
EXTERNAL CONTROL INSTRUCTIONS			
RMCM[3]	Read Memory Controller Mask Registers	3.0	1.9
RMFP[3]	Read Memory File Protect Register	3.0	1.9
SMCM[3]	Set Memory Controller Mask Registers	3.0	1.8
SMFP[3]	Set Memory File Protect Register	3.0	1.8
SMIC[3]	Set Memory Controller Interrupt Cells	3.0	1.8
CIOC[3]	Connect I/O Channel	3.0	1.8
STT	Store Timer Register	3.5	2.5
LDT[2]	Load Timer Register	3.0	1.8
DATA MOVEMENT INSTRUCTIONS, FLOATING-POINT			
FLD	Floating Load	3.0	1.8
DFLD	Double-Precision Floating Load	3.0	1.9
FST	Floating Store	3.5	2.5
DFST	Double-Precision Floating Store	4.0	3.0
LDE	Load Exponent Register	3.0	1.8
STE	Store Exponent Register	3.5	2.5
ARITHMETIC INSTRUCTIONS, FLOATING-POINT			
FAD	Floating Add	3.0	2.7
UFA	Unnormalized Floating Add	3.0	2.5
DFAD	Double-Precision Floating Add	3.0	2.7
DUFA	Double-Precision Unnormalized Floating Add	3.0	2.5
FSB	Floating Subtract	3.0	2.7
UFS	Unnormalized Floating Subtract	3.0	2.5
DFSB	Double-Precision Floating Subtract	3.0	2.7
DUFS	Double-Precision Unnormalized Floating Subtract	3.0	2.5
FMP	Floating Multiply	6.0	5.9
UFM	Unnormalized Floating Multiply	6.0	5.7
DFMP	Double-Precision Floating Multiply	12.0	11.7
DUFM	Double-Precision Unnormalized Floating Multiply	12.0	11.5
FDV	Floating Divide	14.5	14.2
DFDV	Double-Precision Floating Divide	23.5	23.2
FDI	Floating Divide Inverted	14.5	14.2
DFDI	Double-Precision Floating Divide Inverted	23.5	23.2

1 Operations Unit Execution Time Only
2 Functions as NOP is slave mode
3 Causes fault command if executed in slave mode.

Mnemonic Code		Normal Execution Time (μsec) GE-625	GE-635
ARITHMETIC INSTRUCTIONS, FLOATING-POINT			
FNEG	Floating Negate	3.0	2.3
FNO	Floating Normalize	3.0	2.3
ADE	Add to Exponent Register	3.0	1.8
CONTROL INSTRUCTIONS, FLOATING-POINT			
FCMP	Floating Compare	3.0	2.1[1]
DFCMP	Double-Precision Floating Compare	3.0	2.1[1]
FCMG	Floating Compare Magnitude	3.0	2.1[1]
DFCMG	Double-Precision Floating Compare Magnitude	3.0	2.1[1]
FSZN	Floating Set Zero and Negative Compare from Memory	3.0	1.8

[1] Operations Unit Execution Time Only

GE-635 STANDARD CHARACTER SET

Standard Character Set	GE-Internal Machine Code	Octal Code	Hollerith Card Code	Standard Character Set	GE-Internal Machine Code	Octal Code	Hollerith Card Code
0	00 0000	00	0	↑	10 0000	40	11-0
1	00 0001	01	1	J	10 0001	41	11-1
2	00 0010	02	2	K	10 0010	42	11-2
3	00 0011	03	3	L	10 0011	43	11-3
4	00 0100	04	4	M	10 0100	44	11-4
5	00 0101	05	5	N	10 0101	45	11-5
6	00 0110	06	6	O	10 0110	46	11-6
7	00 0111	07	7	P	10 0111	47	11-7
8	00 1000	10	8	Q	10 1000	50	11-8
9	00 1001	11	9	R	10 1001	51	11-9
⌈	00 1010	12	2-8	-	10 1010	52	11
#	00 1011	13	3-8	$	10 1011	53	11-3-8
@	00 1100	14	4-8	*	10 1100	54	11-4-8
:	00 1101	15	5-8	)	10 1101	55	11-5-8
>	00 1110	16	6-8	;	10 1110	56	11-6-8
?	00 1111	17	7-8	'	10 1111	57	11-7-8
ↄ	01 0000	20	(blank)	+	11 0000	60	12-0
A	01 0001	21	12-1	/	11 0001	61	0-1
B	01 0010	22	12-2	S	11 0010	62	0-2
C	01 0011	23	12-3	T	11 0011	63	0-3
D	01 0100	24	12-4	U	11 0100	64	0-4
E	01 0101	25	12-5	V	11 0101	65	0-5
F	01 0110	26	12-6	W	11 0110	66	0-6
G	01 0111	27	12-7	X	11 0111	67	0-7
H	01 1000	30	12-8	Y	11 1000	70	0-8
I	01 1001	31	12-9	Z	11 1001	71	0-9
&	01 1010	32	12	←	11 1010	72	0-2-8
.	01 1011	33	12-3-8	,	11 1011	73	0-3-8
]	01 1100	34	12-4-8	%	11 1100	74	0-4-8
(	01 1101	35	12-5-8	=	11 1101	75	0-5-8
<	01 1110	36	12-6-8	"	11 1110	76	0-6-8
\	01 1111	37	12-7-8	!	11 1111	77	0-7-8